Sistem Monitoring Volume Dan Gas Sampah Menggunakan Metode *Real Time Operating System* (RTOS)

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Tugar Aris Andika Prastiyo Raharjo
NIM: 135150307111021

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

Sistem Monitoring Volume Dan Gas Sampah Menggunakan Metode *Real Time Operating System* (RTOS)

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Tugar Aris Andika Prastiyo Raharjo
NIM: 135150307111021

Skripsi ini telah diuji dan dinyatakan lulus pada
10 July 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Sabriansyah Rizqika Akbar, S.T, M.Eng	Rakhmadhany Primananda, S.T, M.Kom
NIP: 198208092012121004	NIK: 201609804061001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 197105182003121001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 4 Juni 2018

Tugar Aris Andika P.R

NIM: 135150307111021

KATA PENGANTAR

Bagian ini memuat pernyataan resmi untuk menyampaikan rasa terima kasih penulis kepada berbagai pihak yang telah membantu penyelesaian skripsi ini. Nama-nama penerima ucapan terima kasih sebaiknya dituliskan lengkap, termasuk gelar akademik, dan pihak-pihak yang tidak terkait dihindari untuk dituliskan. Bahasa yang digunakan seharusnya mengikuti kaidah bahasa Indonesia yang baku. Kata pengantar boleh diakhiri dengan paragraf yang menyatakan bahwa penulis menerima kritik dan saran untuk pengembangan penelitian selanjutnya. Terakhir, kata pengantar ditutup dengan mencantumkan kota dan tanggal penulisan kata pengantar, lalu diikuti dengan kata “Penulis”.

Malang, 4 Juni 2018

Penulis

Sumenep.aris@gmail.com

ABSTRAK

Sampah merupakan benda-benda yang tidak dipergunakan lagi yang biasanya bersifat kotor dan memiliki bau tidak sedap. Tumpukan sampah pada tempat sampah menghasilkan gas-gas yang dapat mengganggu kesehatan, seperti gas *metana* (CH4), *amonia* (NH3), dan *hidrogen sulfida* (H2S). Jika konsentrasi gas metana terlalu tinggi, maka akan menyebabkan sesak nafas dan kebakaran. Selain itu, volume sampah juga perlu diperhatikan karena jika sampah sudah melebih volume tempat sampah, maka akan menimbulkan kondisi lingkungan yang tidak nyaman. Penelitian ini akan merancang sebuah sistem teknologi yang dapat membantu pemilik tempat sampah mengetahui kondisi tempat sampahnya secara realtime dan kontinyu. Sistem tersebut mengamati volume sampah dan gas-gas yang dihasilkan oleh sensor NH3, CH4, H2S dan infrared E18-D80NK, dengan menggunakan *Real Time Operating System* (RTOS). RTOS digunakan sebagai salah satu tugas pembacaan sensor dan pengiriman yang ditangani oleh mikrokontroller secara *realtime*. Dari hasil 10x pengujian yang dilakukan pada sampah busuk diketahui pembacaan sensor MQ-135 memiliki rata-rata kadar gas amonia yakni 35.71 PPM, MQ-4 memiliki kadar gas metan 293.5 PPM , TGS2602 memiliki kadar9.738 PPM gas hidrogen sufida dan sensor Infrared E18-DN80NK dapat mendeteksi ketinggian sampah dengan keluaran *above threshold* (sampah melebihi batas sensor) dan *below threshold* ketika sampah belum melampaui batas sensor. Jadi kadar gas yang dihasilkan oleh sampah adalah gas metana.

Kata kunci: RTOS, Sensor Gas, Sensor *Infrared*, NodeMCU,Sampah.

ABSTRACT

Rubbish is things that can't be used anymore and it is usually dirty and smelly. A pack of rubbishes in trash bin produce gasses that can give bad impacts to health, like methane (CH4), ammonia (NH3), and hydrogen sulphide (H2S). If the methane concentration is too high, then it will cause asphyxiation and fire. Besides that, rubbish volume also should be noticed because it can cause uncomfortable environment condition if the rubbish has exceeded the volume of trash bin. This research will design a technology system which can help the owner of trash bin to know the trash bin condition in real time and continuously. That system observes the rubbish volume and gases produced by sensors of NH3, CH4, H2S and infrared E18-DN80NK, by using Real Time Operating System (RTOS). RTOS is used as one of the sensor readings and delivery tasks that are handled by the microcontroller in real-time. From the results of 10x testing done on the rotten garbage is known to read the sensor MQ-135 has an average ammonia level of 35.71 PPM, MQ-4 has a methane content of 293.5 PPM, TGS2602 has 9.738 PPM of hydrogen sulphide and Infrared E18-DN80NK sensor can detect the height of the waste by output above the threshold (waste exceeds the sensor limit) and below threshold when the waste has not exceeded the sensor limit. So the level of gas produced by waste is methane gas.

Keywords: *RTOS, Gas Sensor, Infrared Sensor, NodeMCU, Trash.*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar belakang

Sampah merupakan benda-benda yang tidak dipergunakan lagi yang biasanya bersifat kotor dan memiliki bau tidak sedap. Hampir setiap rumah memiliki tempat pembuangan sampah yang merupakan tempat dikumpulkannya sampah yang berasal dari sisa-sisa makanan dan kegiatan lainnya. Tumpukan sampah pada tempat sampah menghasilkan gas-gas yang dapat mengganggu kesehatan, seperti gas *metana* (CH4), *amonia* (NH3), dan *hidrogen sulfida* (H2S). (Radm Robert C. Williams, 2001). Ketiga gas tersebut tidak secara langsung mengganggu kesehatan, namun ketika ketiga konsentrasi gas tersebut terlalu tinggi dan menggantikan oksigen pada suatu ruangan hal tersebut dapat menyebabkan sesak nafas ataupun kebakaran. Gas *amonia, metana* dan *hidrogen sulfida* dapat mengganggu kesehatan pada 0-5 ppm, 200-10000 ppm dan 1.000-5.000 ppm. Selain itu, volume sampah juga perlu diperhatikan karena jika sampah sudah melebih volume tempat sampah, maka sampah akan berserakan dan akan menimbulkan kondisi lingkungan yang tidak nyaman.

Secara psikologi hal tersebut dapat meningkatkan rasa stres dan berkurangnya semangat beraktivitas (Liam Downey, 2011). Melihat masalah yang ditimbulkan oleh sampah ketika volume tempat sampah melebihi kapasitas dan menimbulkan kondisi tidak nyaman pada lingkungan sekitar, penelitian sebelumnya membuat sebuah sistem monitoring untuk pelaporan sampah yang berada di TPS. Sistem ini membantu mengetahui keadaan tempat sampah yang berada di TPS. Sistem yang dibangun menggunakan sensor ultrasonic untuk mendapat data ketinggian sampah dan juga memanfaatkan jaringan radio untuk berkomunkasi data yang diperoleh (Almuchlisin, 2016). Namun pada penelitian ini hanya mendeteksi kepenuhan sampah saja, karena pada umumnya sampah yang telah tertimbun hingga memenuhi kapasitas bak sampah akan menghasilkan bau yang tidak sedap. Biasanya bau tidak sedap tersebut menghasikan gas yang dapat mengganggu kesehatan. Oleh karena itu pada penelitian ini penulis merancang sebuah sistem tidak hanya mengamati volume sampah saja tapi juga mengamati gas gas yang dihasilkan oleh sampah secara real time menggunakan metode *Real Time Operating system* (RTOS). Sistem tersebut mengamati volume sampah dan gas-gas yang dihasilkan menggunakan sensor-sensor yang dipasang pada tempat sampah. Kemudian data hasil bacaan sensor akan ditampikan pada sebuah website sederhana, sehingga dapat diakses dengan mudah oleh perangkat penampil berupa smartphone ataupun komputer yang dapat menjalankan web browser yang memiliki koneksi internet. Penulis mengharapkan dengan sistem yang dirancang dapat mengatasi permasalahan-permasalahan yang ditimbulkan oleh sampah.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah disampaikan dapat disusun rumusan masalah sebagai berikut:

1. Bagaimana cara membaca kondisi tempat sampah, yaitu volume tempat sampah, gas CH4, gas H2S, dan gas NH3 yang dihasilkan oleh sampah?
2. Bagaimana cara mengimplementasikan metode *Real Time Operating System* (RTOS) pada mikrokontroller untuk membaca kondisi tempat sampah?
3. Bagaimana performa sistem dalam mengamati kondisi tempat sampah yang diukur dari pengiriman data ?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disampaikan sebelumnya dapat disusun tujuan-tujuan yang ingin dicapai, yaitu meliputi:

1. Membaca kondisi tempat sampah, yaitu volume tempat sampah, gas CH4, gas H2S dan gas NH3 yang dihasilkan oleh sampah.
2. Mengimplementasikan metode *Real Time Operating System* (RTOS) pada mikrokontroleler untuk membaca kondisi tempat sampah.
3. Mengukur performa sistem dalam mengamati kondisi tempat sampah yang diukur dari pengiriman data.

1.4 Manfaat

Manfaat yang dapat diambil dari skripsi ini meliputi:

1. Memungkinkan pemilik tempat sampah mengamati kondisi tempat sampahnya secara realtime.
2. Mencegah permasalahan-permasalahan kesehatan dan psikologi yang ditimbulkan oleh sampah.

1.5 Batasan masalah

Skripsi ini mempunyai beberapa batasan-batasan permasalahan, antara lain:

1. Web *server* di-deploy hanya pada jaringan lokal.
2. Jumlah tempat sampah dibatasi sebanyak 1 buah.
3. Menggunakan sampah basah dan kering.

1.6 Sistematika pembahasan

Untuk memahami lebih jelas pembahasan proposal skripsi ini, dilakukan dengan cara mengelompokkan materi menjadi beberapa subbab dengan sistematika pembahasan sebagai berikut :

BAB I Pendahuluan

Bab pendahuluan memuat penjelasan mengenai skripsi ini yang berjudul "*Sistem Monitoring Volume Dan Gas Sampah Menggunakan Metode Real Time Operating System (RTOS)*" dan mengapa hal tersebut perlu dikerjakan.

BAB II Landasan Kepustakaan

Bab landasan kepustakaan memuat uraian dan pembahasan tentang dasar teori dan kajian pustaka.

BAB III Metodologi

Bab metodologi memuat uraian dan pembahasan tentang metode dan langkah-langkah penelitian yang digunakan pada skripsi ini.

BAB IV Rekayasa Kebutuhan

Membahas semua yang dibutuhkan oleh sistem. Mulai dari kebutuhan perangkat keras dan lunak, kebutuhan fungsional dan non fungsinonal.

BAB V Perancangan dan Implementasi

Menjelaskan proses sistem perancangan alat Trashmonitoring hingga implementasi perangkat keras dan perangkat lunak.

BAB VI Pengujian dan Analisis

Membahas pengujian yang telah dilakukan pada sistem dan melakukan analisa terhadap hasil pengujian yang telah dilakukan.

BAB VII Penutup

Membahas kesimpulan dari hasil penelitian yang dilakukan, serta memberikan saran yang dapat digunakan untuk pengembangan serupa untuk kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang metode penelitian yang digunakan, kajian pustaka yang menjelaskan secara umum penelitian-penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan, perbedaan skripsi tersebut terhadap penelitian terdahulu yang dituliskan.

2.1 Kajian pustaka

Tinjauan pustaka memuat pembahasan mengenai penelitian-penelitian terdahulu yang berkaitan dengan topik skripsi. Tujuan dari melakukan kajian pustaka adalah untuk mengkaji hasil penelitian sebelumnya dan dijadikan sebagai dasar dalam pelaksanaan penelitian.

Kajian pustaka pertama yang dijadikan sumber adalah Agung Leona Suparlin pada tahun 2017-2018 yang berjudul "Implementasi *System Real Time* Untuk *Monitoring* Pencahayaan Suhu Dan Kelembaban Pada Tanaman Stroberi". Pada penelitian ini bertujuan untuk memonitoring tanaman stroberi dengan 3 sensor menggunakan metode RTOS yaitu pencahayaan, suhu dan kelembaban tanah (Agung Leona Suparlin, 2017). Dan kajian pustaka kedua yang dijadikan sumber adalah Almichlisin pada tahun 2016-2017 yang berjudul "Perancangan dan Implementasi Sistem monitoring Untuk Pelaporan Sampah Berbasis Teknologi Embeded". Pada penelitian ini bertujuan untuk memonitoring kepenuhan sampah pada TPS dengan menggunakan sensor ultrasonic dan jaringan radio sebagai komunikasi data yang diperoleh. Berikut ini adalah perbedaan kajian pustaka dengan penelitian sebelumnya pada tabel 2.1 dibawah ini (Almuchlisin, 2016).

Tabel 2.1 Kajian Pustaka

No.	Nama Penulis, Tahun, dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	Agung Leona Suparlin [2017-2018] Real Time Operating System On Arduino	Menerapkan Real Time Operating System pada Arduino	Menerapkan RTOS pada sebuah sistem monitoring tanaman stroberi dengan menggunakan 3 sensor yaitu pencahayaan, suhu dan kelembaban tanah	Menerapkan RTOS pada sebuah sistem monitoring sampah dengan menggunakan 4 sensor yaitu Infrared, MQ4, MQ135, TGS2602

No.	Nama Penulis, Tahun, dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
2.	Almuclisin[2016-2017] Perancangan dan Implementasi Sistem monitoring Untuk Pelaporan Sampah Berbasis Teknologi Embeded	Menerapkan Sampah sebagai Objek Penelitian untuk mengetahui kepenuhan sampah pada TPS	Meneliti Keadaan kepenuhan sampah pada TPS menggunakan sensor *untrasonic* dan jarinagn radio sebagai komunkasi data yang diperoleh.	Meneliti keadaan volume dan gas yang dihasilkan oleh sampah menggunakan sensor infrared, MQ135, MQ4 dan TGS2602
				Menerapkan metode *Real Time Operating system*

2.2 Dasar teori

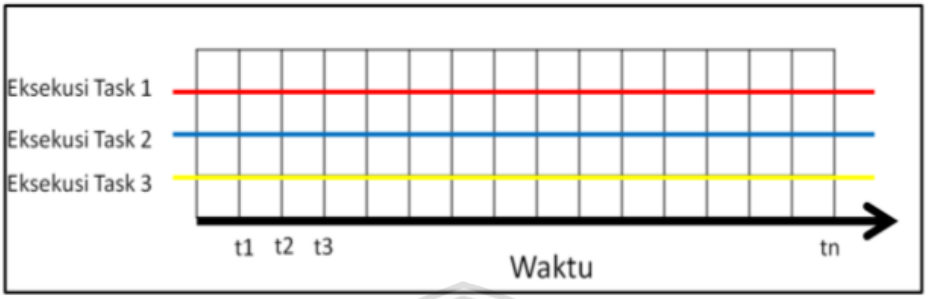
Dasar teori memuat pembahasan mengenai teori yang diperlukan untuk menyusun penelitian.

2.2.1 Real Time Operating System (RTOS)

RTOS merupakan metode yang digunakan untuk menyelesaikan skripsi ini. RTOS adalah perangkat lunak yang berperan sebagai penjadwalan waktu eksekusis setiap program. Waktu eksekusi dibagi sedemikian rupa sehingga waktu pergantian antar program dapat diperediksi dengan tepat. RTOS berbeda dengan sistem operasi seperti pada umumnya yang terdapat pada komputer. Sistem operasi pada komputer akan bekerja sesaat ketika *power* masuk ke komputer, kemudian program berjalan. Lain halnya dengan RTOS, sistem operasi ini dijalankan secara otomatis oleh program kernel. Ketika sistem dinyalakan, maka *kernel* akan menyala terlebih dahulu kemudian baru menyalakan *Real-Time Operating System*. Tugas utama dari RTOS ialah mengatur sumber daya yang ada meliputi processor, memori/register, serta melakukan komunikasi dan sinkronisasi. RTOS memiliki beberapa konsep dasar diantaranya *multitasking* dan scheduling.

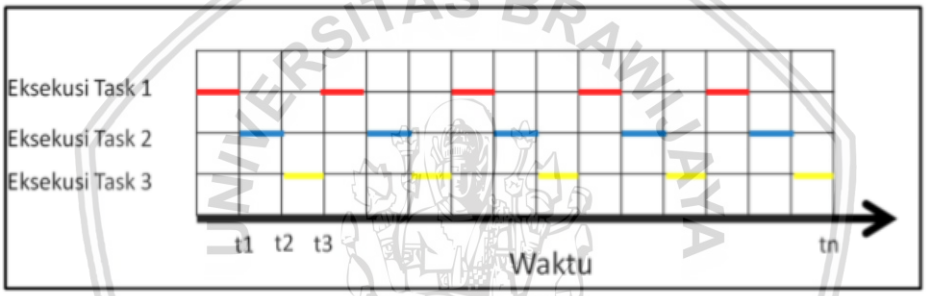
1. Multitasking adalah kemampuan sistem operasi dalam menjalankan banyak *task* atau *thread* dalam rentan waktu tertentu. Konsep *multitasking* dari sistem ini ialah dapat memudahkan desain sebuah aplikasi yang komplek menjadi sederhana dan dapat diatur, Bagian-bagian kecil dari sebuah aplikasi dapat dengan mudah

diuji coba, ditelusuri, dan didaur ulang untuk kebutuhan lain. Detail alur dari aplikasi serta pengaturan waktu eksekusi yang kompleks dapat dihilangkan dari kode aplikasi. Hal tersebut sudah menjadi tanggung jawab dari sistem operasi itu sendiri.. Konsep *multitasking* dan konkuren dapat dilihat pada Gambar 2.1 dan Gambar 2.2.



Gambar 2.1 Konsep *Multitasking*

Sumber: (diadaptasi dari Jatmiko, 2015)



Gambar 2.2 Konsep konkurensi

Sumber: (diadaptasi dari Jatmiko, 2015)

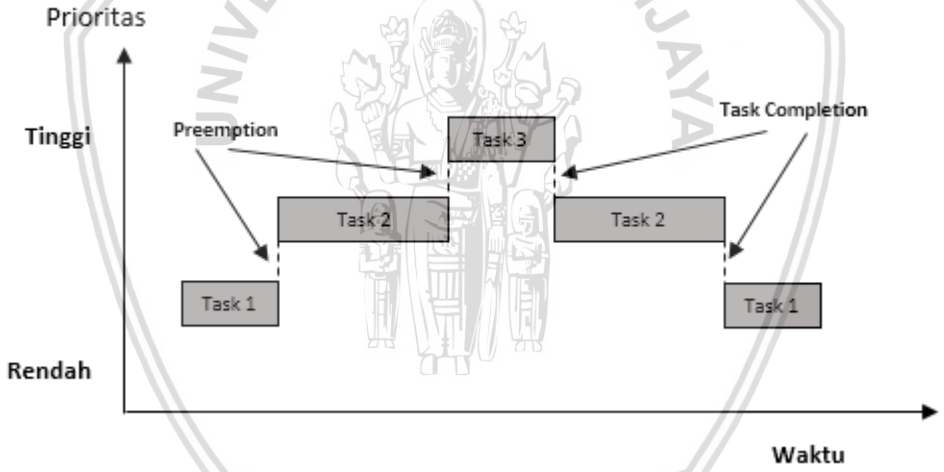
Pada Gambar 2.1 menunjukkan konsep *multitasking* bekerja. Tiga *task* berjalan dalam waktu yang sama dan waktu eksekusi yang sama. Akan tetapi pada memori konvensional, konsep *multitasking* belum dapat dijalankan dikarenakan sumber daya prosesor dan memori yang terbatas sehinga membatasi kinerja dari konsep *multitasking*. Pada gambar 2.2 menunjukan *task-task* yang ada, di eksekusi secara bergantian. Hal ini dinamakan dengan konkurensi. Konsep konkurensi adalah konsep dimana sistem operasi sudah mengetahui *task* mana saja yang akan dijalankan dalam rentan waktu tertentu. Selanjutnya sistem operasi membagi rentan waktu yang ada untuk mengeksekusi beberapa *task*. Kemudian sistem operasi mengatur kapan waktu eksekusi setiap *task* dan menyimpan posisi dari setiap *task* pada memori, sehingga ketika prosesor menjalankan suatu *task,* prosesor akan mengambil state terakhir dari *task*. Proses perpindahan antara eksekusi *task* dilakukan dengan cepat dan konkuren sehingga seolah - olah terlihat procesor melakukan beberapa *task* dalam rentan waktu bersamaan. *Embedded* sistem dapat menggunakan *RTOS* untuk menjalankan beberapa *task* secara konkuren seolah-olah dapat bekerja secara *multitasking.*

2. Scheduling merupakan konsep pembagian waktu eksekusi *task-task* yang ada. Untuk melakukan *scheduling* dibutuhkan adanya sebuah *scheduler*. *Scheduler*

berfungsi untuk menentukan waktu eksekusi suatu *task,* lama eksekusi *task*, waktu suatu *task* ditunda (*suspend*), dan waktu *task* dijalankan kembali (*resume*). Pada penetuan waktu eksekusi *task* diperlukan suatu algoritma *scheduling*. Didalam *kernel* RTOS terdapat 2 algoritma *scheduling* diantaranya.

- *Preemptive Priority-Based Scheduling*

RTOS menggunakan algoritma *Preemptive Priority-Based Scheduling* sebagai algoritma default untuk *scheduling*. Algoritma ini mengatur pada setiap waktu *task* dengan *prioritas* paling tinggi akan dieksekusi. *Task* tersebut akan didahulukan dari *task* maupun yang juga sudah siap untuk dieksekusi. Gambar 2.3 menunjukkan bagaimana algoritma *Preemptive Priority-Based Scheduling* bekerja. *Task* dengan *prioritas* tinggi akan dieksekusi, sedangkan *task* lain akan ditunda dulu hingga *task* dengan *prioritas* lebih tinggi selesai dieksekusi. Pada RTOS, penentuan *prioritas* pada setiap *task* dilakukan pada saat pertama kali dibuat. Perubahan *prioritas* juga perlu memperhatikan pembagian *prioritas* yang sudah ada sehingga tidak menimbulkan *priority inversion*, *deadlock*, maupun kegagalan sistem.

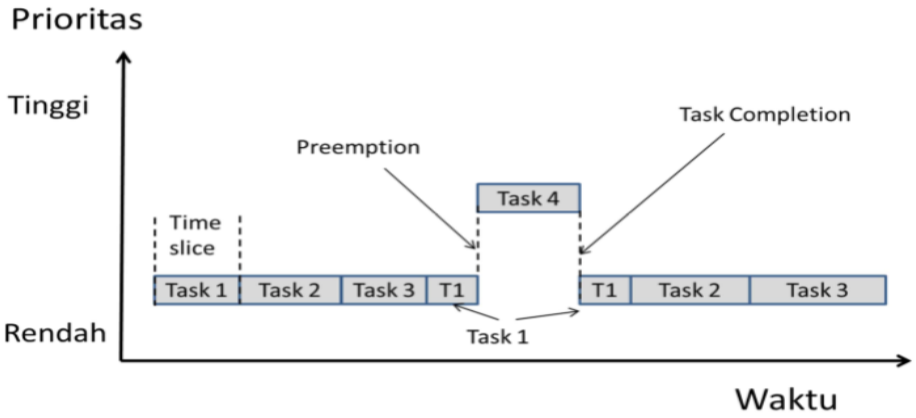


Gambar 2.3 Algoritma *preemptive prioritt-based schaduling*

Sumber: (diadaptasi dari Jatmiko, 2015)

- *Round-robin scheduling*

Algoritma ini mengatur setiap task memperoleh jatah waktu eksekusi yang sama. Round robin pada dasarnya kurang cocok digunakan untuk real-time sistem karena tidak memberikan variasi prioritas pada setiap task. Padahal pada real-time sistem setiap task akan dieksekusi berdasarkan variasi prioritas yang ada. Round robin scheduling menggunakan time slicing untuk membagi mendapatkan alokasi waktu yang sama pada setiap task.



Gambar 2.4 Kombinasi algoritma round dan preemtivr scheadulling

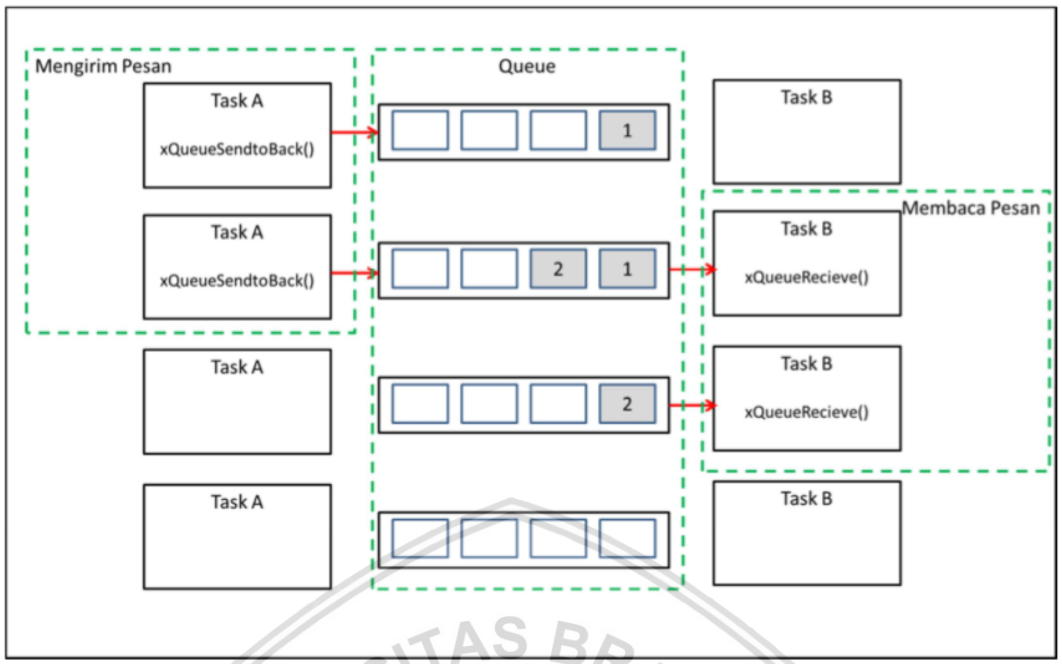
Sumber: (diadaptasi dari Jatmiko, 2015)

Sesuai pada Gambar 2.4 round robin akan membagi waktu eksekusi task menggunakan time slice. Terdapat counter yang akan menghitung waktu eksekusi setiap task. Ketika waktu sudah habis maka eksekusi akan dilanjutkan pada task berikutnya. Setiap task yang selesai dieksekusi akan ditempatkan pada akhir round. Kernel RTOS melakukan modifikasi terhadap algoritma Round Robin Scheduling yaitu mengkombinasikannya dengan preemptive scheduling. Modifikasi yang dilakukan ialah jika terdapat task lain dengan prioritas lebih tinggi tiba-tiba muncul maka counter yang menghitung task yang sedang dieksekusi akan disimpan (eksekusi task1 di-intterupt oleh task 4, prioritas task 4 lebih tinggi dari task 1, task 2, dan task 3). Kemudian task dengan prioritas tinggi akan dieksekusi, setelah selesai eksekusinya, kernel akan kembali mengeksekusi task yang sebelumnya telah di-intterupt melanjutkan waktu yang sudah dicatat counter sebelumnya (Wisnu Jatmiko, 2015).

2.2.2 *Free Real Time Operating System* (FreeRTOS)

FreeRTOS adalah salah satu Real Time Operating System (RTOS) yang dapat dijalankan untuk embedded system. Sistem operasi ini dapat dipasang pada mikrokontroler. Sistem tertanam biasanya menggunakan satu atau lebih mikrokontroler sebagai integrated circuit (IC) untuk menjalankan fungsi tertentu. Mikrokontroler terdiri dari prosessor sekaligus memori baik ROM, Flash, maupun RAM, serta beberapa komponen penunjang seperti transistor, clock, resistor, dan lain-lain. FreeRTOS menyediakan beberapa fitur utama suatu Real Time Operating System. Fitur tersebut diantaranya real time scheduling functionality, inter-task communication, timing, dan synchronization primitives.

FreeRTOS memiliki fitur queues dimana fitur ini digunakan untuk mengirim pesan antar task dan antara task dengan interrupt. Pesan yang dikirim dari satu task ke task lain dimasukkan dalam satu queue dengan Algoritma First In First Out (FIFO). Algoritma FIFO berperan ketika pesan yang dilewatkan antar task lebih dari satu. Gambar 2.5 menjelaskan bagaimana queue dalam RTOS bekerja.



Gambar 2.5 Alur kerja antrian

Untuk menjalankan aplikasi RTOS di freeRTOS ini membutuhkan alokasi memori *scheduler* 236 *byte, Queue 76 byte,* dan *task* sebesar 64 *byte.*

2.2.3 Mikrokontroler arduino mega

Arduino mega platform merupakan sebuah single board computer yang mudah digunakan dan mumpuni yang diminati oleh banyak pasar profesional dan hobi. Arduino mega memiliki 54 pin digital *input / output,* 15 diantatranya dapat digunakan sebagai output PWM, 16 buah analog *input,* 4 UART. Arduino Mega juga dilengkapi 16 Mhz kristal isolator, sebuah *port* USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset.* Arduino Mega bersifat open-source yang mana hardware memiliki harga cukup rendah dan development software bersifat gratis. Arduino mega Project mulai dikembangkan di Itali untuk kebutuhan hardware dengan harga murah untuk desain interaksi. Berikut salah satu produk hardware dan development software Arduino mega:

Gambar 2.6 Arduino mega

Sumber : (https://forum.arduino.cc)

Pada gambar 2.6 dapat dilihat bahwa terdapat 2 bagian pada program arduino mega, yaitu setup() dan loop(). Setup() adalah bagian program yang dijalankan hanya sekali saat pertama kali hardware hidup dan biasanya digunakan untuk kebutuhan konfigurasi awal. Sedangkan, loop() adalah bagian program yang dijalankan berulang-ulang setelah setup() dan biasanya digunakan untuk menjalankan pekerjaan utama hardware (Durfee, 2011). Spesifikasi arduino mega dapat dilihat pada *tabel* 2.2 (ArduinoMega_DataSheet,2018).

Tabel 2.2 Spesifikasi arduino mega

Spesifikasi Arduino Mega	
Microcontroler	*ATmega2560*
Operating Voltage	5V
Input Voltage (Recomended)	7-12V
Input Voltage (Limits)	6-20V
Digital I/O Pins	54 (*of which14 periode PWM output*)
Analog input Pins	16
DC current per I/O Pin	20 mA
DC current for 3.3V Pin	50 mA
Flash Memory	256 KB *of which 8 KB use by bootloader*
SRAM	8 KB
EEPROM	4 KB
Clock speed	16 MHz
LED_BUILDTIN	13
Panjang	101.52 mm
Lebar	53.3 mm
Berat	37 g

2.2.4 Mikrokontroler Nodemcu

Pada gambar 2.7 menjelaskan perangkat keras NodeMCU. NodeMCU adalah sebuah platform IoT yang bersifat *opensource.* Yang terdiri dari perangkat keras berupa Sistem *On Chip* dari ESP8266, Modul WiFi nodeMCU ini menggunakan bahasa pemrograman *scripting* Lua yang juga dilengkapi dengan GPIO, ADC, UART dan PWM. NodeMCU merupakan mikroprosesor open soucre yang berjalan pada Esp8266 WiFi SoC dari Espresssif Sistem dan perangkat keras yang berbasis ESP12 module sehingga mendukung koneksi WiFi. NodeMCU dapat dihubungkan dengan komputer menggunakan kabel USB dan dapat diprogam dengan mudah karena NodeMCU kompatible dengan dua program editor, yaitu LUA dan Arduino IDE

serta dapat diprogram dengan bahasa pemrograman C atau C++. NodeMCU memiliki keunggulan sendiri dibanding dengan mikrokontroler lainnya, yaitu dari segi bentuk yang lebih kecil serta memiliki fungsi yang lebih lengkap dibanding Arduino Uno seperti kemampuan untuk terhubung ke jaringan *wireless* tanpa harus ada perangkat tambahan. Prangkat ini dapat aktif pada tegangan 3.3V (Schwastz, 2017).

Gambar 2.7 Nodemcu

Sumber : (http://lampatronics.com/product/nodemcu/)

2.2.5 Sensor infrared E18-D80NK

Pada gambar 2.8 menjelaskan perangkat keras Infrared E18-D80NK. Sensor infrared E18-D80NK ialah sensor yang telah memiliki transmiter dan receiver yang dikemas menjadi satu. Sensor ini memiliki 3 pin yakni diantaranya kabel merah sebagai VCC, kabel hijau sebagai GND, dan kabel kuning sebagai D0. Sensor ini deteksi Objek yang mampu mendeteksi objek dalam jarak 3-50 cm dengan konsumsi tegangan maksimum 5V DC. Jarak deteksi 3-50cm dapat diatur sesuai keperluan dengan memutar potensiometer pada bagian belakangnya. Untuk memperpanjang range dengan cara memutar searah jarum jam potensio (CW) dan sebaliknya akan memperkecil range pengukuran pada sensor. Pada kepala Proximity switch ini terdapat sepasang Transmitter dan Receiver untuk mendeteksi objek/halangan. Prinsip kerja dari sensor E18-D80NK ialah untuk mendeteksi ada atau tidaknya suatu objek. Bila objek berada didepan sensor dan dapat terjangkau oleh sensor maka output rangkain sensor akan berlogika “1” atau “high” yang berarti objek “ada”. Sebaliknya jika objek sensor bernilai “0” atau “low” maka “tidak ada”. Peneliti menggukana Sensor infrared E18-D80NK untuk mengambil data sensor ketinggian yang berada ditempat sampah, karena sensor tersebut memiliki kalibrasi yang akurat, jarak deteksi 3 samapai 80 cm, serta harganya yang terjangkau (Carl K. Chang, 2016).

Gambar 2.8 Infrared E18-D80NK

Sumber : (http://lampatronics.com/product/infrared E18-D80NK/)

2.2.6 Sensor MQ-135

Pada gambar 2.9 menjelaskan perangkat keras sensor MQ-135. Sensor MQ-135 adalah sensor yang dapat mendeteksi beberapa jenis gas yakni NH3,Nox,alkohol Benzene,CO2 dan lain lain (MQ-135_Datasheet,2016). Prinsip kerja dari sensor MQ-135 ialah membaca konduktifitas gas yang berada disekitar sensor, sesuai dengan pembacaan resistansi di pin analog. Sensor gas MQ-135 dapat bekerja ketika diberi tegangan input sebesar 5V, serta memiliki jangkauan deteksi gas 10-10000 pmm dengan sensitifitas yang berbeda pada masing masing gasnya. Sensor MQ-135 memiliki 4 pin diantaranya A0, D0, GND dan VCC. Sensor ini membaca nilai gas secara analog sehingga pin output yang digunakan pada sensor ini ialah A0. VCC berperan sebagai tegangan input, GND bertindak sebagai tegangan output, dan D0 sebagai pin output digital. Peneliti menggukan sensor MQ-135 untuk mengambil data gas amonia yang dihasilkan oleh bau busuk sampah misalnya, sisa-sisa makanan, tumbuhan mati dan kotoran hewan. Sensor ini dipilih karena dapat mendeteksi gas amaonia yang dihasilkan oleh bau busuk sampah tersebut. Selain itu sensor ini memiliki fitur kalibrasi yang luamayan akurat dan harganya pun terjangkau (Aynur Unal, 2016).

Gambar 2.9 Sensor MQ135

Sumber : (http://lampatronics.com/product/mq-135-gas-sensor/)

2.2.7 Sensor MQ4

Pada gambar 2.10 menjelaskan perangkat keras sesor MQ4. Sensor MQ-4 ialah sensor yang dapat mendeteksi beberapa jenis dianataranya CH4, Natural Gas, LNG, *avoid the noise of alcochol*, *cooking fumes, cigarette smoke* (MQ-4_Datasheet,2018). Sensor MQ-4 memiliki 4 pin diantaranya A0, D0, GND dan VCC. Sensor ini membaca nilai gas secara analog sehingga pin output yang digunakan pada sensor ini ialah A0. VCC berperan sebagai tegangan input, GND bertindak sebagai tegangan output, dan D0 sebagai pin output digital. Prinsip kerja dari sensor MQ-4 ialah membaca konduktifitas gas yang berada disekitar sensor, sesuai dengan pembacaan resistansi di pin analog. Sensor gas MQ-4 dapat bekerja ketika diberi tegangan input sebesar 5V, serta memiliki jangkauan deteksi gas 200-10000 ppm dengan sensitifitas yang berbeda pada masing-masing gas. Peneliti menggukan sensor ini untuk mengambil data gas metan yang dihasilkan oleh bau busuk sampah misalnya seperti kotoran hewan dan manusia, sisa makanan, dan sisa-sisa tumbuhan mati. Sensor ini memiliki fitur kalibrasi yang akurat dan harganya pun terjangkau (Aynur Unal, 2016).

Gambar 2.10 Sensor MQ4

Sumber : (http://lampatronics.com/product/mq-4-gas-sensor/)

2.2.8 Sensor TGS2602

Pada gambar 2.11 menjelaskan perangkat keras sensor TGS2602. Sensor TGS2602 adalah sensor yang dapat mendeteksi beberapa jenis sensor diantaranya kadar gas hidrogen sulfida dan amonia yang dihasikan oleh limbah dikantor ataupun lingkungan rumah. Sensor TGS2602 memiliki 4 pin diantaranya A0, D0, GND dan VCC. Sensor ini membaca nilai gas secara analog sehingga pin output yang digunakan pada sensor ini ialah A0. VCC berperan sebagai tegangan input, GND bertindak sebagai tegangan output, dan D0 sebagai pin output digital. Prinsip kerja dari sensor TGS2602 ialah membaca konduktifitas gas yang berada disekitar sensor, sesuai dengan pembacaan resistansi di pin analog. Sensor gas TGS2602 dapat bekerja ketika diberi tegangan input sebesar 5V, serta memiliki jangkauan deteksi gas 1-100 pmm dengan sensitifitas yang berbeda pada masing masing gas. Peneliti menggunakan sensor ini untuk mengambil data sensor gas hydrogen sulfida yang dihasilkan oleh bau busuk sampah misalnya seperti telur busuk. Selain itu sensor ini memiliki fitur kalibrasi yang akurat akan tetapi harganya lumayan mahal (Albastaki, 2018).

Gambar 2.11 Sensor TGS2602

Sumber : : (http://lampatronics.com/product/tgs2602-gas-sensor/)

2.2.9 HTML

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web (willard, 2006).

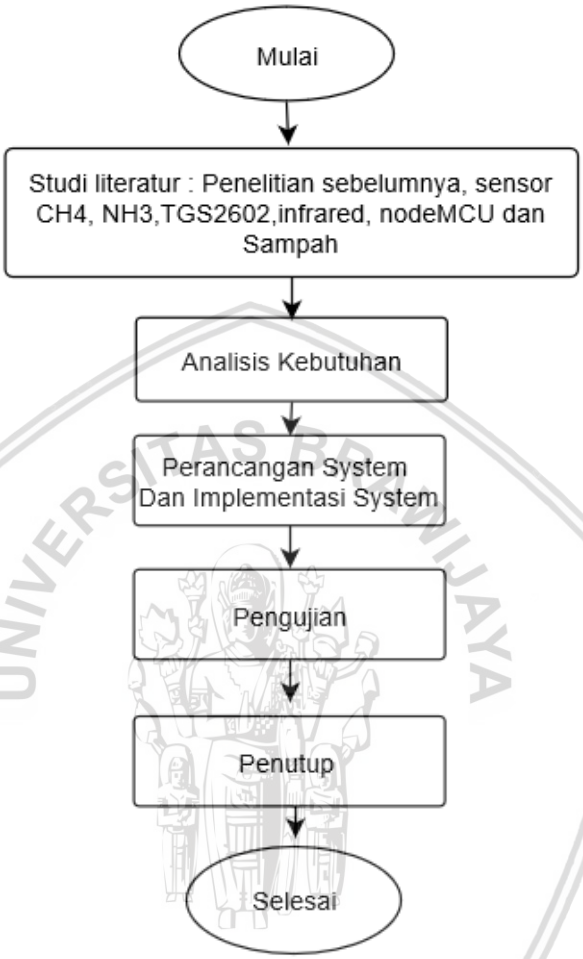
2.2.10 Sampah

Sampah merupakan benda-benda yang tidak dipergunakan lagi yang biasanya bersifat kotor dan memiliki bau tidak sedap. Hampir setiap rumah memiliki tempat pembuangan sampah yang merupakan tempat dikumpulkannya sampah yang berasal dari sisa-sisa makanan dan kegiatan lainnya.

Tumpukan sampah pada tempat sampah menghasilkan gas-gas yang dapat mengganggu kesehatan, seperti gas metana (CH4), ammonia (NH3), dan hidrogen sulfida (H2S) (Radm Robert C. Williams, 2001). Gas metana tidak secara langsung mengganggu kesehatan, namun jika konsentrasi gas tersebut terlalu tinggi dan menggantikan oksigen pada suatu ruangan dapat menyebabkan sesak nafas. Gas ammonia dan hidrogen sulfida dapat mengganggu kesehatan pada 0,5-1 ppm dan 1.000-5.000 ppm. Ketiga gas tersebut terutama metana juga dapat menyebabkan kebakaran atau ledakan pada konsentrasi tertentu, terlebih jika berada pada ruangan tertutup. Selain itu, volume sampah juga perlu diperhatikan karena jika sampah sudah melebih volume tempat sampah, maka sampah akan berserakan dan akan menimbulkan kondisi lingkungan yang tidak nyaman. Secara psikologi hal tersebut dapat meningkatkan rasa stres dan berkurangnya semangat beraktivitas (Liam Downey, 2011).

BAB 3 METODOLOGI

Bab metodologi memuat uraian dan pembahasan tentang metode dan langkah-langkah penelitian yang digunakan pada diagram alir Gambar 3.1.



Gambar 3.1 Diagram alir metode penelitian

Makna dari metodologi penelitian dapat dilihat dari dua sudut pandang. Pertama, dari pandangan umum dia bisa berarti sebuah cara sistematik untuk menyelesaikan masalah penelitian. Dalam hal ini dia juga dapat merupakan kumpulan cara (metode) yang lebih spesifik dalam penyelesaian masalah. Kedua, metodologi penelitian dapat dipahami sebagai sebuah ilmu untuk mempelajari bagaimana sebuah penelitian dilakukan secara sistematik. Dalam ilmu ini kita mempelajari berbagai langkah yang umumnya digunakan oleh peneliti ketika mempelajari masalah penelitian beserta alasan-alasan logis di belakangnya. Oleh karena itu di dalam pembahasan metodologi penelitian, yang dibicarakan tidak hanya metode, teknik, atau langkah-langkah yang digunakan dalam sebuah penelitian tetapi juga logika di balik metode, teknik, atau langkah-langkah tersebut sesuai dengan konteks penelitiannya masing-masing. Dalam hal ini perlu dijelaskan mengapa sebuah metode atau teknik dipilih.

3.1 Studi literatur

Pada bagian ini dibahas mengenai dasar teori yang mendukung penelitian Sistem Monitoring Volume Dan Gas Yang Dihasilkan Oleh Sampah menggunakan metode Real Time Operating System (RTOS). Teori-teori yang digunakan dalam tahapan studi literatur ini berasal dari buku, jurnal, *website* resmi, artikel, serta *e-book*. Berikut merupakan dasar teori yang digunakan sebagai bahan studi:

1. Real Time Operating System (RTOS)

   Mempelajari tentang konsep dasar dari RTOS, cara kerja dari metode RTOS, serta kelebihan dan kekurangan dari metode ini dalam melakukan penjadawalan sebuah task.

2. Mikorkntroler arduino mega

   Mempelajari terkait konfigurasi penggunaan mikrokontroler arduino mega sebagai pengelolah sistem baik dari segi konfigurasi pin secara perangkat keras maupun dari segi penggunaan algoritma dan logika yang tepat agar sesuai dengan hasil yang diharapkan.

3. Mikrokontroler nodemcu

   Mempelajari terkait konfigurasi mikrokontroler nodemcu sebagai modul Wifi

4. Sensor Infrared E18-D80NK

   Mempelajari terkait spesifikasi sensor dan cara konfigurasi pin sensor untuk dapat membaca volume tempat sampah yang diteliti.

5. Sensor MQ135

   Mempelajari terkait spesifikasi sensor dan cara konfigurasi pin sensor untuk dapat membaca nilai gas amonia yang dihasilkan sampah yang diteliti.

6. Sensor MQ4

   Mempelajari terkait spesifikasi sensor dan cara konfigurasi pin sensor untuk dapat membaca nilai gas metana yang dihasilkan sampah yang diteliti.

7. Sensor TGS2602

   Mempelajari terkait spesifikasi sensor dan cara konfigurasi pin sensor untuk dapat membaca nilai gas amonia yang dihasilkan sampah yang diteliti.

8. HTML

   Mempelajari tentang kode pemograman html dalam pembuatan web sederhana yang digunakan untuk menampilkan data dari sensor sensor yang digunakan.

9. Sampah

   Mempelajari terkait gas gas yang dihasilkan oleh sampah busuk.

3.2 Analisis kebutuhan

Analisis kebutuhan diperlukan untuk menganalisis apa saja yang dibutuhkan oleh sistem pada penelitian yang dilakukan, sehingga dapat sistem dapat melakukan sesuai dengan yang diharapkan. Berikut beberapa kebutuhan fungsional yang dibutuhkan oleh sistem dalam penelitian ini :

1. Sensor jarak dapat membaca volume sampah pada bak sampah
2. Sensor gas dapat membaca kadar amonia, metan dan hidrogen sulfida
3. Data dari sensor dapat diolah menggukan metode *Real Time Operating System* (RTOS).
4. Sistem dapat menampilkan data hasil dari volume dan kadar gas yang dihasilkan oleh sensor pada website.

3.2.1 Kebutuhan perangkat keras

Berikut kebutuhan perangkat keras yang digunakan untuk membuat sistem pada penelitian ini :

1. Mikrokontroller Arduino Mega
2. Sensor Gas MQ135
3. Sensor Gas MQ4
4. Sensor Gas TGS2602
5. Sensor Infrared E18-D80NK
6. NodeMCU
7. Laptop
8. Mega Sensor Shield
9. Power Bank

3.2.2 Kebutuhan perangkat lunak

Berikut beberapa kebutuhan perangkat lunak yang digunakan oleh sistem pada penelitian ini iyalah berupa Arduino IDE 1.6.4 , Arduino_FreeRTOS.h, Semaphor.h, dan arduino library.

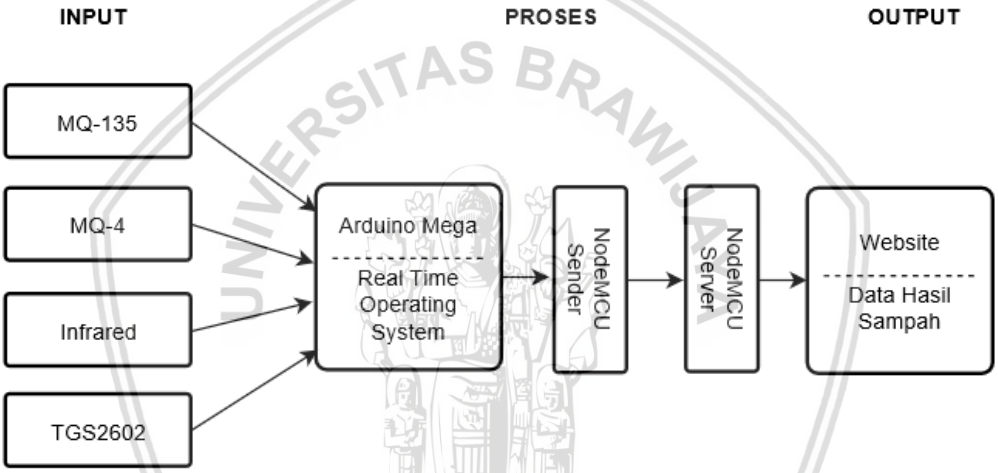
3.3 Perancangan dan implementasi sistem

Setelah identifikasi masalah dilakukan, tahapan selanjutnya ialah perancangan dan implementasi sistem. Perancangan dan implementasi sistem terdiri dari beberapa perancangan dan implementasi fungsional yaitu perancangan dan implementasi fitur-fitur perangkat keras berbasis arduino mega dan NodeMCU yang dapat berfungsi sebagaimana yang telah dipaparkan ditahap identifikasi masalah, yaitu membaca sensor dan mengirimkan datanya ke perangkat monitor melalui NodeMCU *sender* ke NodeMCU *server* yang kemudian ditampilka pada website. Selain perangkat keras, perangkat lunak juga dijalankan. Perangkat lunak

tersebut mengimplementasikan *Real Time Operating System* (RTOS). Pada implementasi juga menampilkan gambar sistem yang sudah dibuat serta potongan-potongan bahasa pemrograman yang digunakan.

3.3.1 Perancangan perangkat keras

Pada tahap perancangan perangkat keras sistem yang akan dibuat yakni berupa blok diagram. Terlihat dari Gambar 3.2 terdapat 3 (tiga) bagian utama yakni Input, Process, dan Output. Pada bagian Input terdapat 4 (empat) buah sensor yang berfungsi untuk membaca kadar gas amonia,kadar gas metana, kadar gas hidrogen sulfida dan ketinggian sampah, lalu pada bagian Process terdapat Arduino Mega yang berfungsi sebagai mikrokontroller pengolah data dari sensor untuk dilakukan klasifikasi menggunakan *Real Time Operating System (RTOS)*, dan pada bagian Output akan menampilkan hasil nilai dari kadar gas yang dihasilkan oleh sampah tersebut melalui Website.



Gambar 3.2 Blok diagram sistem

3.3.2 Perancangan perangkat lunak

Perancangan perangkat lunak pada subbab ini yakni berupa penerapan *Real Time Operating System* (RTOS) pada mikrokontroller yang sesuai dengan tujuan utama dari pembuatan sistem ini. Sistem akan bekerja ketika ada pembacaan data dari sensor yang kemudian dilanjutkan dengan penjadwalan pada tiap sensor infrared, NH3, CH4, dan H2S dengan menggunakan *Real Time Operating System* (RTOS) agar data sensor tersebut terschedule dengan baik dan konsisten secara waktu maupun secara task applikasi yang dilakukan, setelah itu hasil dari nilai pembacaan tiap sensor ditampilkan pada website.

3.4 Pengujian

Pengujian dilakukan untuk mengukur sejauh mana hasil yang diperoleh. Selain itu, untuk mengetahui seperti apa performa perangkat lunak dan mekanisme komunikasi yang dirancang dalam mencapai tujuan yang dijelaskan pada identifikasi masalah. Pengujian yang dilakukan antara lain:

1. Pengujian sensor gas.
    - Sensor mq135.
    - Sensor mq4.
    - Sensor tgs2602.
2. Pengujian sensor infrared.
3. Pengujian *Real Time Operating System* (RTOS) pada mikrokontroler.
    - Eksekusi task berdasarkan prioritas.
    - Waktu eksekusi task pada sistem.
    - Perbandingan wkatu eksekusi task menggunakan metode RTOS dengan tanpa menggunakan metode RTOS.
4. Pengujian pengiriman data.
    - Pengiriman data dari arduino ke nodemcu sender.
    - Pengiriman data dari nodemcu sender ke nodemcu server.

3.5 Penutup

Dalam bab ini menjelaskan tentang penutup dari skripsi ini. Setelah melakukan seluruh alur penelitian, kemudian melakukan penarikan kesimpulan dari apa yang telah diteliti. Kesimpulan dapat diambil dari hasil pengujian dari penulisan. Bab ini juga akan menjelaskan saran untuk penelitan selanjutnya agar dapat mendapatkan pembaharuan dari penelitian ini.

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan dijelaskan secara rinci terkait gambaran umum sistem, analisis kebutuhan fungsional, kebutuhan perangkat keras, kebutuhan perangkat lunak dan batasan desain sistem.

4.1 Gamabran umum sistem

Sistem ini bertujuan untuk membantu pemilik tempat sampah mengetahui kondisi tempat sampahnya. Bantuan teknologi ini diperlukan, karena dapat memungkinkan pengawasan secara realtime dan kontinyu. Sistem tersebut mengamati volume sampah dan gas-gas yang dihasilkannya menggunakan sensor-sensor yang dipasang pada tempat sampah. Data hasil bacaan sensor kemudian dikirim ke web *server* secara wireles, sehingga dapat diakses oleh perangkat penampil. Protokol tersebut digunakan, karena memiliki message overhead yang kecil dan sifat pengiriman yang realtime. Tugas pembacaan dan pengiriman tersebut ditangani oleh mikrokontroller secara realtime menggunakan *Real Time Operating System (RTOS)*. Perangkat penampil dapat berupa smartphone atau komputer yang dapat menjalankan web browser dan memiliki koneksi internet. Data-data sensor yang telah dikirim akan ditampilkan pada sebuah website, sehingga data tersebut dapat diakses melalui perangkat penampil dengan mudah agar dapat mengetahui kondisi tempat sampah.

4.2 Analisis kebutuhan sistem

Analisis kebutuhan sistem dilakukan untuk menggali semua kebutuhan yang diperlukan untuk Sistem Monitoring ini. Dalam melakukan anlisis kebutuhan sistem terdiri atas beberapa kebutuhan yang perlu dijabarkan yakni kebutuhan fungsional dan kebutuhan non fungsional, dimana kebutuhan non fungsional terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak. Hasil dan pembahasan dapat diletakkan dengan kemungkinan berikut:

4.2.1 Kebutuhan fungsional

Berikut ini adalah kebutuhan fungsional sistem yang harus dilakukan :

1. Sensor gas dapat membaca kadar amonia dalam sampah.
   Sensor gas MQ135 bertugas untuk membaca dan mengakuisisi nilai kadar amonia dalam sampah yang dideteksi. Nilai yang dihasilkan berupa konsentrasi kadar amonia dengan satuan PPM.
2. Sensor gas dapat membaca kadar metana dalam sampah.
   Sensor gas MQ-4 bertugas untuk membaca dan mengakuisisi nilai kadar gas metan dalam sampah yang dideteksi. Nilai yang dihasilkan berupa konsentrasi kadar metana dengan satuan PPM.
3. Sensor gas dapat membaca kadar amonia dalam sampah.

Sensor gas TGS2602 bertugas untuk membaca dan mengakuisisi nilai kadar hidrogen sulfida dalam sampah yang dideteksi. Nilai yang dihasilkan berupa konsentrasi kadar hidrogen sulfida dengan satuan PPM.

4. Sensor Infrared E18-D80NK dapat membaca ketinggian sampah pada tempat sampah.
   Sensor infrared E18-D80NK bertugas untuk membaca ketinggian sampah pada tempat sampah. Nilai yang dihasilkan berupa sebuah kondisi 0 yang berarti *Above* dan 1 yang berarti *Below*.
5. Mikrokontroler nodemcu *sender* dan nodemcu *server*
   Mikrokontroller NodeMCU *sender* bertugas untuk menerima dan mengirimkan data dari arduino yang telah telah dischedulingoleh FreeRTOS kemudian dikirim ke NodeMCU *server*. Nilai data sensor pengiriman data dari NodeMCU *sender* akan di tampilkan pada website sederhana yang dapat diakses melalui perangkat penampil.

4.2.2 Kebutuhan non fungsional

Kebutuhan non fungsional dari sistem ini ialah kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dijelaskan dibawah ini.

4.2.2.1 Kebutuhan perangkat keras

Untuk mendukung implementasi pembuatan sistem ini maka diperlukan beberapa alat dari perangkat keras yang dijelaskan dibawah ini.

1. Arduino mega

   Mikrokontroller yang digunakan pada sistem monitoring sampah ini adalah mikrokontroler rduino mega. Pengguna menggunakan arduino mega dikarenakan memiliki daya yang minimal dan bentuk fisik yang kecil sehingga dapat diterapkan didaerah ekstrim. Pemilihan arduino mega juga dikarenakan memiliki pin yang lebih lengkap dari pada tipe arduino lainnya untuk kebutuhan sensor yang akan dipakai oleh pengguna. Selain itu arduino mega juga memiliki kapasitas memori yang lebih banyak. Arduino mega memakai voltase rendah yaitu 5 Volt. Arduino mega juga memiliki regulator voltase dimana jika voltase yang masuk belebih maka akan dibuang menjadi panas untuk menghindari kerusakan *board*. Selain itu, Arduino mega memiliki *library* yang dapat membantu secara langsung misalnya dalam pembacaan sensor MQ-135, CH4, Infrared E18-D80NK dan H2S, memakai bahasa C/C++ dalam pemrogramannya, Arduino mega ini juga memiliki banyak referensi, sehingga sangat membantu dalam proses perangkaian sensor pada sistem ini. Mikrokontroller ini dijual bebas dengan harga yang terjangkau sehingga mudah didapat.

2. Kabel Jumper

   Kabel *Jumper* yang dipakai pada Tash Monitoring *Sender* ini antara lain bertipe *male-to-male* dan *male-to-female*. Jumper berguna untuk menghubungkan antar komponen-komponen pada node tersebut.

3. Sensor Infrared E18-D80NK

   Sensor infrared E18-D80NK digunakan untuk mengambil data sensor ketinggian yang berada ditempat sampah. Sensor ini dipilih karena memiliki fitur kalibrasi yang sangat akurat. Sensor ini juga memiliki nilai beli yang lumayan murah.

4. Sensor MQ-135

   Sensor MQ-135 digunakan untuk mengambil data sensor gas amonia yang dihasilkan oleh bau busuk sampah misalnya kotoran hewan dan manusia, sisa makanan, dan sisa-sisa tumbuhan mati. Sensor ini dipilih karena kandungan yang terdapat pada sampah ketika busuk rata-rata ialah amonia. Selain itu sensor ini memiliki fitur kalibrasi yang luamyan akurat dan harganya pun terjangkau.

5. Sensor MQ-4

   Sensor MQ-4 digunakan untuk mengambil data sensor gas metan yang dihasilkan oleh bau busuk sampah misalnya seperti kotoran hewan dan manusia, sisa makanan, dan sisa-sisa tumbuhan mati. Sensor ini dipilih karena kandungan yang terdapat pada sampah ketika busuk rata-rata ialah gas metan. Selain itu sensor ini memiliki fitur kalibrasi yang akurat dan harganya pun terjangkau.

6. Sensor TGS2602

   Sensor TGS2602 digunakan untuk mengambil data sensor gas hydrogen sulfida yang dihasilkan oleh bau busuk sampah misalnya seperti telur busuk. Sensor ini dipilih karena kandungan yang terdapat pada sampah ketika busuk rata-rata ialah ammonia. Selain itu sensor ini memiliki fitur kalibrasi yang akurat dan harganya pun mahal.

7. Mega Sensor Shield

   Mega Sensor Shield digunakan untuk mempermudah pengguna untuk merancang sitem di arduino mega tanpa menggunakan projek boar tambahan lagi.

8. NodeMCU *Sender* dan *Server*

   NodeMCU *Sender* bertindak sebagai menerima data dari hasil akuisis semua sensor yang kemudian diteruskan ke NodeMCU *server* yang mana data tersebut akan di tampilkan ke website.

9. Mobile hotspot

   Mobile hotsport berfungsi sebagai router penghubung antara NodeMCU *Sender* dan NodeMCU *Server* dalam pengiriman data.

4.2.2.2 Kebutuhan perangkat lunak

1. Arduino Mega IDE

   Pemrograman pada mikrokontroller Arduino mega dilakukan di *software* Arduino mega Integrated Development Environment (IDE) menggunakan bahasa C/C++. Bahasa pemrograman Arduino mega IDE ini menggunakan bahasa pemrograman Java dan juga dilengkapi banyak *library* baik dari bawaan arduino maupun yang dapat diunduh untuk memudahkan proses pemrograman di Arduino mega.

   Tampilannya yang sederhana, ukuran software yang luamyan kecil serta *user-friendly* yang sangat cocok untuk digunakan para pemula dalam mempelajari pemrograman Arduino mega ini. Terdapat fitur serial monitor yang tanpa harus menggunakan LCD pada Arduino untuk melihat berbagai pertukaran data antara mikrokontroller dengan port serial yang terhubung dengannya. Arduino mega IDE *compatible* dengan berbagai macam board Arduino.

2. Arduino mega FreeRTOS Library

   Library ini digunakan untuk menscheduling data hasil akuisis sensor gas dan ketinggian pada tempat sampah yaitu Infrahmerah, NH3, CH4, dan H2S sehingga dapat diproses oleh mikrokontroller Arduino mega.

3. Arduino NodeMCU *sender* dan sever Library

   Library NodeMCU *sender* ini digunakan untuk proses pengiriman data hasil akusisi sensor yang telah dischedulingoleh *Real Time Operating System* (RTOS) yang kemudian dikirim ke nodeMCU *server* yang kemudian dikirim ke website.

4. HTML

   HTML adalah sebuah bahasa pemograman yang digunakan untuk membuat sebuah website untuk menampilkan nilai hasil kadar gas dan ketinggian pada sampah.

4.3 Batasan sistem

Dalam pembuatan sistem monitoring ini ada beberapa batasan sistem agar ruang lingkup pembahasan, perancangan, maupun pengimplementasiannya tidak terlalu luas. Batasan desain sistem ini sebagai berikut :

1. Web server di-deploy hanya pada jaringan lokal.
2. Jumlah tempat sampah dibatasi sebanyak 1 buah.
3. Menggukan sampah basah dan kering.
4. Menggunakan 2 nodemcu yang bertindak sebagai sender dan server.
5. Nilai output sensor dalam keadaan tempat kosong tidak bisa 0.

BAB 5 PEMBAHASAN

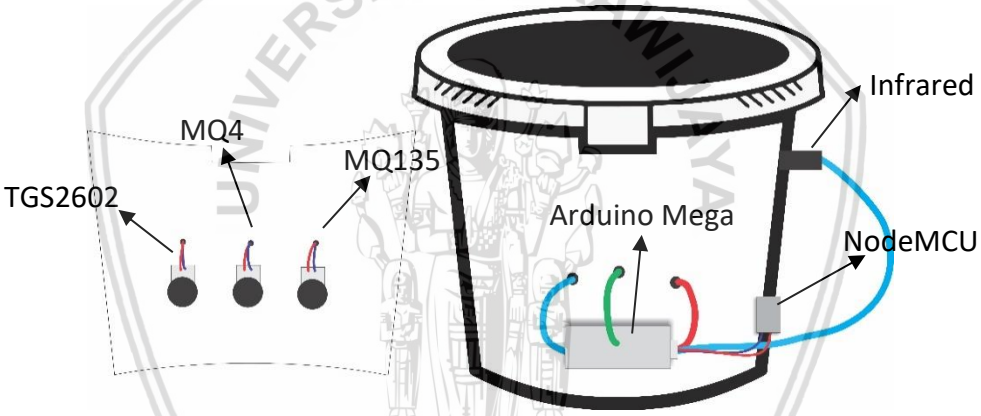
Pada bab ini menjelaskan tentang mekanisme perancangan sistem dari penelitian dan implementasi sistem yang meliputi implementasi perangkat keras,perangkat lunak, dan implementasi pendukung sistem lainnya.

5.1 Perancangan sistem

Proses perancangan sistem dibagi menjadi beberapa tahapan yang dimulai dari gambaran sistem secara umum. Kemudian dilanjutkan dengan perancangan perangkat keras dan perangkat lunak serta perancangan peralatan pendukung sistem ini.

5.1.1 Perancangan *prototype* alat monitoring sampah

Untuk melakukan desain *prototype* dari sistem Monitoring sampah ini perlu diperhatikan peletakan tiap tiap komponen. Pembuatan desain sistem ini menggunakan tempat sampah yang ditunjukkan pada Gambar 5.1.



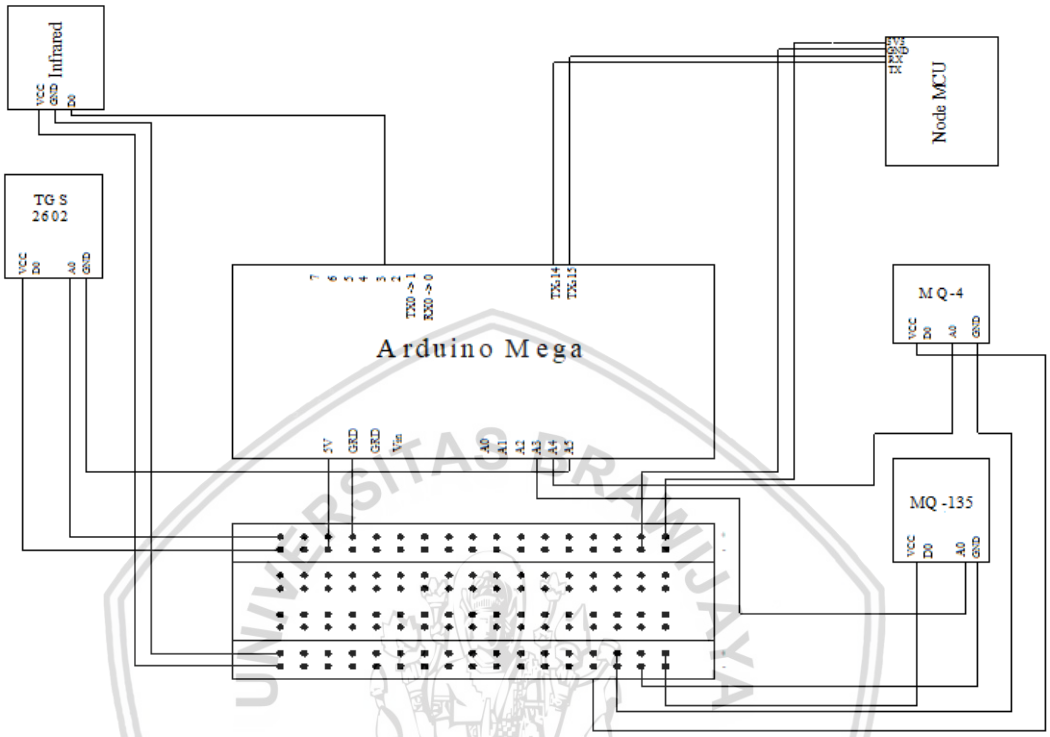
Gambar 5.1 Desain *prototype* alat monitoring sampah

Posisi sensor MQ-4 , Sensor MQ-135 , dan sensor TGS2602 diletakkan di samping bawah tempat samapah, selain gas yang akan dibaca juga ada volume tempat sampah melalui tinggi sampah yang dideteksi oleh sensor infrared E18-D80NK,peletakan sensor tersebut yaitu dibagian atas tempat sampah, kemudian penempatan mikrokontroler nodemcu di samping arduino mega, serta mikrokontroller arduino mega yang diletakkan disampaing *prototype* tempat sampah.

5.1.2 Perancangan perangkat keras

Perancangan perangkat keras dikerjakan berdasarkan analisis kebutuhan serta spesifikasi dari masing masing perangkat keras agar dapat membangun sistem yang diharapkan. Berdasarkan Gambar 3.2 yang berupa blok diagram sistem maka perangkat keras ini menjelaskan secara detail skematik pin-pin antar tiap komponen perangkat keras, yaitu sensor gas MQ-4, sensor MQ-135, sensor TGS2602, dan sensor infrared E18-D80NK yang merupakan inputan dari mickrokontroller Arduino mega sebagai pengeloh data sehingga hasil olahan data

tersebut akan dikirim melalui nodeMCU *sender* menuju nodeMCU *server* yang kemudian data tersebut akan ditampilkan pada website. Skematik diagram perancangan perangkat keras sistem monitoring sampah ini ditunjukan pada Gambar 5.2



Gambar 5.2 Diagram skematik sistem

Pada gambar 5.2 diagram skematik akan menunjukkan koneksi antar pin nodeMCU *sender* dengan arduino mega pada Tabel 5.1. NodeMCU *sender* ini memiliki 4 pin diantaranya GND, Vcc, RX, TX. Dimana Vcc dihubungkan dengan pin 5V pada arduino mega. Pin Rx pada nodeMCU dihubungkan ke Pin Tx pada arduino mega begitupun sebaliknya. Dan pin GND dihubunkan dengan pin GND pada arduino mega. NodeMCU *sender* ini berfungsi menerima data dari hasil akuisis semua sensor yang kemudian diteruskan ke NodeMCU *server* yang mana data tersebut akan di tampilkan ke website.

Tabel 5.1 Keterangan koneksi pin nodemcu dengan arduino mega

Pin nodeMCU *Sender*	Pin Arduino mega
Vcc	5V
GND	GND
TX	RX
RX	TX

Tabel 5.2 menunjukkan koneksi antar pin sensor gas MQ135 dengan Arduino mega. Pada sistem ini menggunakan module sensor gas MQ135 yang mempunyai 4 pin diantaranya A0, D0, GND dan VCC. Sensor ini membaca nilai gas secara analog sehingga pin output pada sensor ini yang digunakan ialah A0 yang dihubungkan dengan pin A0 pada arduino mega, kemudian pin VCC dihubungkan dengan pin 5V pada arduino mega dan pin GND dihubungkan dengan pin GND pada aduino. Sensor gas MQ135 merupakan salah satu jenis sensor yang bersifat resistif maka diperlukan rangkaian pembagi tegangan agar dapat dibaca oleh mikrokontroler, oleh karena itu pin A0 pada sensor MQ135 tidak serta merta langsung dihubungkan dengan pin A0 pada arduino melainkan harus dibagi dulu dengan sebuah resistor.

Tabel 5.2 Keterangan koneksi pin sensor MQ135 dengan arduino mega

Sensor MQ135	Pin Arduino mega
Vcc	Vcc
GND	GND
A0	A3

Tabel 5.3 menunjukkan koneksi antar pin sensor gas MQ-4 dengan Arduino mega. Pada sistem ini menggunakan module sensor gas MQ-4 yang memiliki 4 pin diantaranya A0, D0, GND dan VCC. Sensor ini membaca nilai gas secara analog sehingga pin output pada sensor ini yang digunakan ialah A0 yang dihubungkan dengan pin A0 pada arduino mega, kemudian pin VCC dihubungkan dengan pin 5V pada arduino mega dan pin GND dihubungkan dengan pin GND pada aduino. Sensor gas CH4 merupakan salah satu jenis sensor yang bersifat resistif maka diperlukan rangkaian pembagi tegangan agar dapat dibaca oleh mikrokontroler, oleh karena itu pin A0 pada sensor CH4 tidak serta merta langsung dihubungkan dengan pin A0 pada arduino melainkan harus dibagi dulu dengan sebuah resistor.

Tabel 5.3 Keterangan koneksi pin sensor MQ4 dengan arduino mega

Sensor MQ-4	Pin Arduino mega
Vcc	Vcc
GND	GND
A0	A4

Tabel 5.4 menunjukkan koneksi antar pin sensor gas TGS2602 dengan Arduino mega. Pada sistem ini menggunakan module sensor gas TGS2602 yang memiliki 4 pin diantaranya A0, D0, GND dan VCC. Sensor ini membaca nilai gas secara analog sehingga pin output pada sensor ini yang digunakan ialah A0 yang dihubungkan dengan pin A0 pada arduino mega, kemudian pin VCC dihubungkan dengan pin 5V pada arduino mega dan pin GND dihubungkan dengan pin GND pada aduino. Sensor gas TGS2602 merupakan salah satu jenis sensor yang bersifat

resistif maka diperlukan rangkaian pembagi tegangan agar dapat dibaca oleh mikrokontroler, oleh karena itu pin A0 pada sensor TGS2602 tidak serta merta langsung dihubungkan dengan pin A0 pada arduino melainkan harus dibagi dulu dengan sebuah resistor

Tabel 5.4 Keteranga koneksi pin sensor TGS2602 dengan arduino mega

Sensor TGS2602	Pin Arduino mega
Vcc	Vcc
GND	GND
A0	A5

Tabel 5.5 menunjukkan koneksi antar pin sensor infrared E18-D80NK dengan Arduino mega . Pada sensor ini memiliki 3 pin diantaranya D0, GND dan VCC. Sensor ini membaca volume tempat sampah melalui tinggi sampah yang memiliki nilai keluaran digital sehingga pin output pada sensor ini yang digunakan ialah D0 yang dihubungkan dengan pin D3 pada arduino mega, kemudian pin VCC dihubungkan dengan pin 5V pada arduino mega dan pin GND dihubungkan dengan pin GND pada aduino.

Tabel 5.5 Keterangan koneksi pin Infrared dengan arduino mega

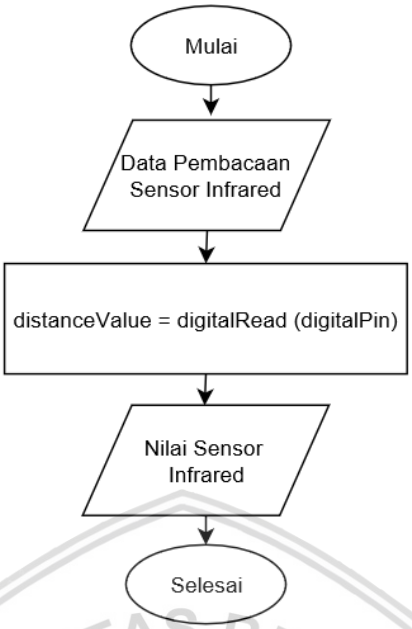
Sensor Infrared E18-D80NK	Pin Arduino mega
Vcc (kabel warna merah)	Vcc
GND (kabel warna hijau)	GND
D0 (kabel warna kuning)	D5

5.1.3 Perancangan perangkat lunak

Pada subbab pembahasan perancangan perangkat lunak ini dibagi menjadi 2, yakni pembahasan perancangan perangkat lunak pada mikrokontroller untuk pengambilan data sensor yang diolah serta pembahasan perancangan perangkat lunak untuk melakukan pengiriman data dari nodeMCU *sender* ke nodeMCU *server*.

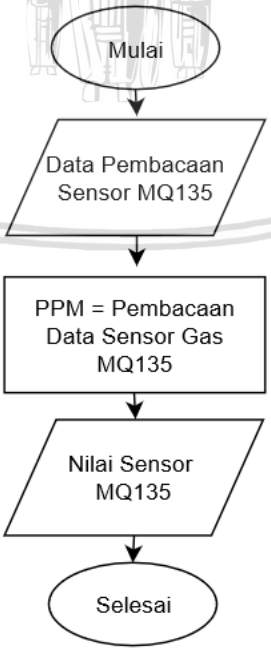
5.1.3.1 Perancangan pembacaan data sensor Infrared, MQ135, MQ4, TGS2602

Pada gambar 5.3 menujukkan Diagram Alir Pembacaan Sensor Nilai Infrared. Dimana nilai data sensor Infrared ini merupakan nilai data sensor dari Infrared E18-D80NK, dimana nilai sensor infrared ini mengeluarkan nilai digital yang berlogika ‘1’ dan ‘0’. Dimana logika ‘1’ direpresentasikan ‘high’ atau objek ‘ada’ dan logika ‘0’ direpresentasikan ‘low’ atau objek ‘tidak ada’. Nilai dari data sensor Infrared digunakan untuk menentukan ketinggian sampah pada tempat sempah.



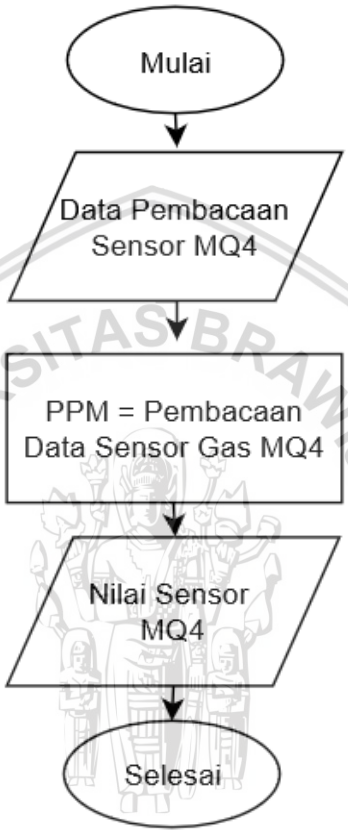
Gambar 5.3 Diagram alir pembacaan sensor Infrared

Pada gambar 5.4 menujukkan Diagram Alir Pembacaan Sensor Nilai MQ135, Dimana nilai data sensor MQ135 ini merupakan salah satu jenis sensor yang bersifat resistif maka diperlukan rangkaian pembagi tegangan agar dapat dibaca oleh mikrokontroler. nilai pembagi tegangan tersebut digunakan untuk mencari nilai resefif yang kemudian digunakan untuk mencari nilai satuan ppm pada sensor gas CH4.



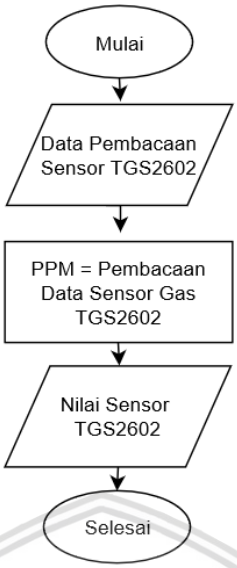
Gambar 5.4 Diagram alir pembacaan sensor MQ135

Pada gambar 5.5 menujukkan Diagram Alir Pembacaan Sensor Nilai MQ4, Dimana nilai data sensor MQ4 ini merupakan salah satu jenis sensor yang bersifat resistif maka diperlukan rangkaian pembagi tegangan agar dapat dibaca oleh mikrokontroler, nilai pembagi tegangan tersebut digunakan untuk mencari nilai resefif yang kemudian digunakan untuk mencari nilai satuan ppm pada sensor gas CH4.



Gambar 5.5 Digram alir pembacaan sensor MQ4

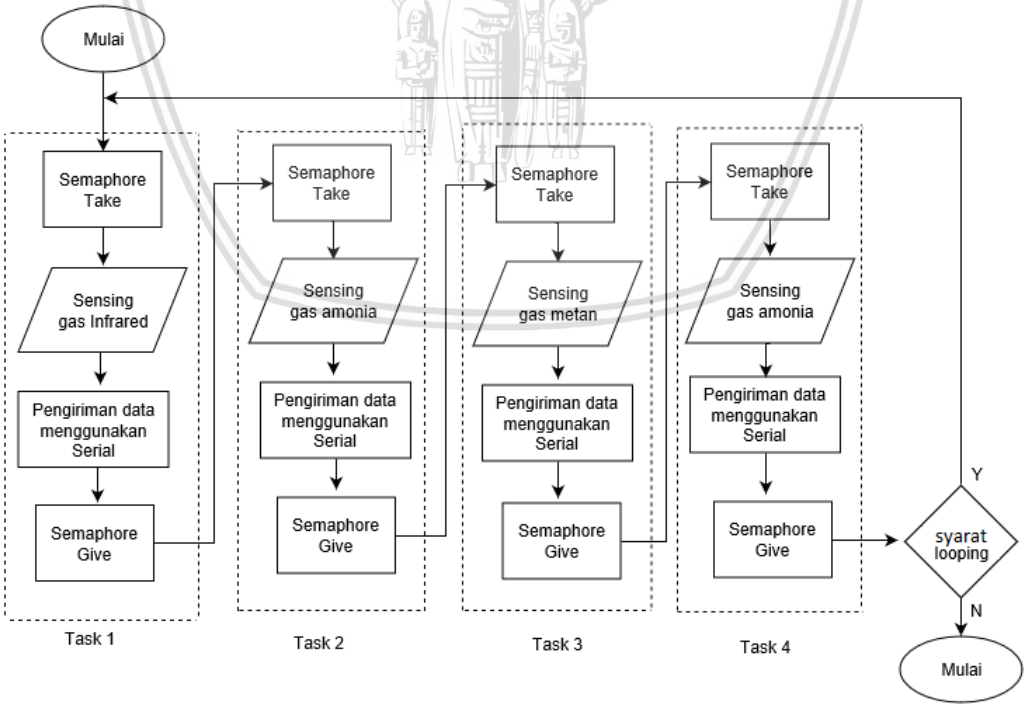
Pada gambar 5.6 menujukkan Diagram Alir Pembacaan Sensor Nilai TGS2602, Dimana nilai data sensor TGS2602 ini salah satu jenis sensor yang bersifat resistif maka diperlukan rangkaian pembagi tegangan agar dapat dibaca oleh mikrokontroler, nilai pembagi tegangan tersebut digunakan untuk mencari nilai resefif yang kemudian digunakan untuk mencari nilai satuan ppm pada sensor gas H2S.



Gambar 5.6 Diagram alir pembacaan sensor TGS2602

5.1.3.2 Perancangan RTOS

Pada proses perancangan RTOS terdiri dari beberapa tahapan yaitu menentukan berapa *task* yang akan dibuat, menentukan tugas tiap *task*, mentukan *priority* tiap *task*, dan mengatur *delay* tiap *task*nya. Diagram alir perancangan RTOS dapat dilihat pada Gambar 5.7.



Gambar 5.7 Diagram alir perancangan RTOS

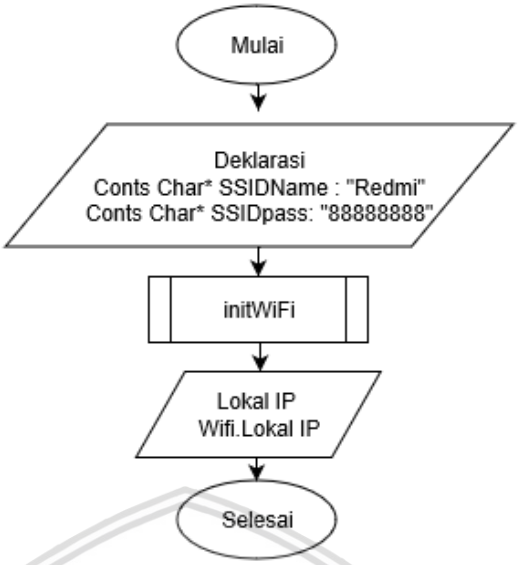
Seperti Gambar 5.7, sistem akan dibagi menjadi 4 *task* dimana tiap *task* memiliki tugasnya masing-masing tiap *task* akan menggunakan sumber yang sama yaitu *Semaphore*. *Semaphore* pada sistem adalah suatu fungsi yang bertugas agar *task* tidak dapat di *interrupt* oleh *task* lain selama *task* tersebut masih dieksekusi**.** Dengan penggunaan *Semaphore* tiap *task* akan memakai sumber daya secara bergantian agar dapat berjalan secara konkuren. Ketika sistem dijalankan, *task* dengan *prioritas* terbesar akan melakukan pengambilan *Semaphore* dan mulai mengeksekusi *task* hingga selesai. Setelah eksekusi *task* tersebut selesai, *task* akan melepaskan *Semaphore* untuk digunakan oleh *task* dengan *prioritas* terbesar berikutnya. Pada persyaratan *looping* sistem akan kembali melakukan perulangan jika semua *task* berjalan tanpa melebihi *deadline* yang telah ditentukan tetapi apabila *task* yang di eksekusi melebihi *deadline* yang telah ditentukan maka sistem tidak akan berjalan dan mengalami kegagalan sistem. Untuk pembagian tugas, *prioritas* serta *deadline*nya lebih jelasnya dapat dilihat pada Tabel 5.5.

Tabel 5.5 Pembagian Tugas, prioritas dan *deadline* pada sistem

No	Task	Tugas	Priorotas	Deadline
1	NH3	Melakukan sensing gas amonia pada sampah dan dikirim ke nodeMCU *Sender*	2	1000 ms
2	CH4	Melakukan sensing gas metan pada sampah dan dikirim ke nodeMCU *Sender*	2	1000 ms
3	H2S	Melakukan sensing gas hidrogen sulfida pada sampah dan dikirim ke nodeMCU *Sender*	2	1000 ms
4	Infrared	Melakukan sensing gas amonia pada sampah dan dikirim ke nodeMCU *Sender*	2	1000 ms

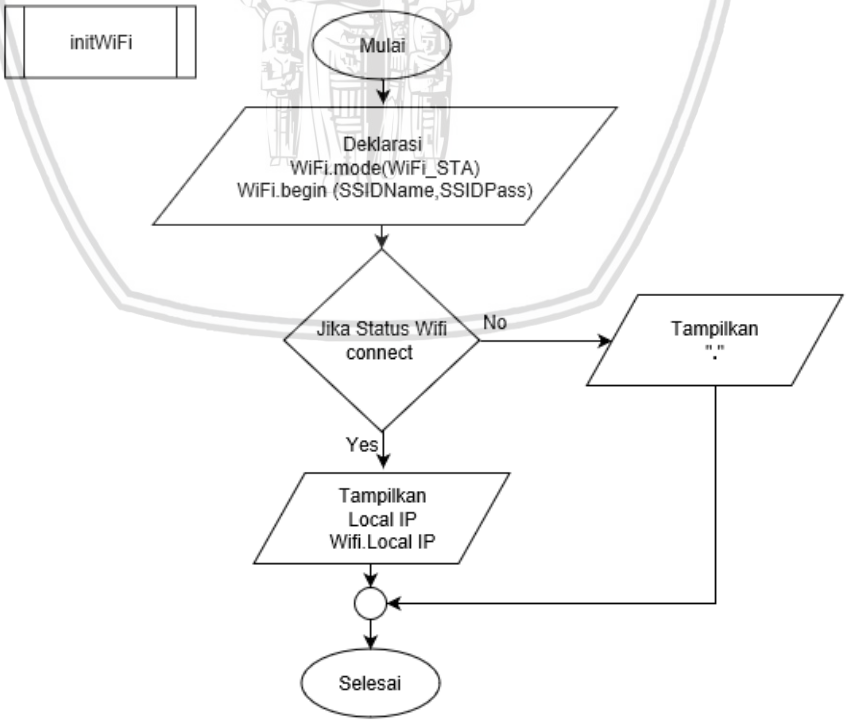
5.1.3.3 Perancangan koneksi wifi

Pada Gambar 5.8 diagram alir perancangan koneksi wifi diatas masukan awal pada tahapan ini adalah pendeklarasian nama wifi yakni : “Redmi” dan pasword dari dari wifi tersebut ialah “88888888”. Setelah pendeklarasian nama dan pasword pada diagram alir tersebut terdapat fungsi initWiFi () yang didalamnya terdapat sebuah konsidi yang mempengaruhi koneksi wifi. Adapun penejelasan dari fungsi yang ditunjukan dan dijelaskan pada diagram berikut.



Gambar 5.8 Diagram Alir Koneksi WiFi

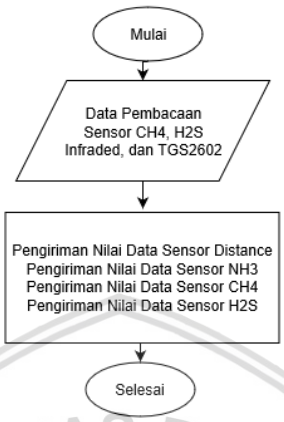
Gambar 5.9 diagram alir fungsi initWiFi adalah fungsi initWiFi yang terdapat pada gambar 5.8 perancangan koneksi WiFi, fungsi ini diawali dengan masukan pendeklarasian WiFi_STA , nama WiFi, dan pasword WiFi. Dimana didalam fungsi initWiFi terdapat sebuah kondisi, jika Status WiFi terhubung maka akan menampilkan Local IP, dan jika status WiFi tidak terhubung maka akan menampilkan ".".



Gambar 5.9 Diagram alir fungsi initWiFi ()

5.1.3.4 Perancangan pengiriman data sensor dari arduino ke nodmcu sender

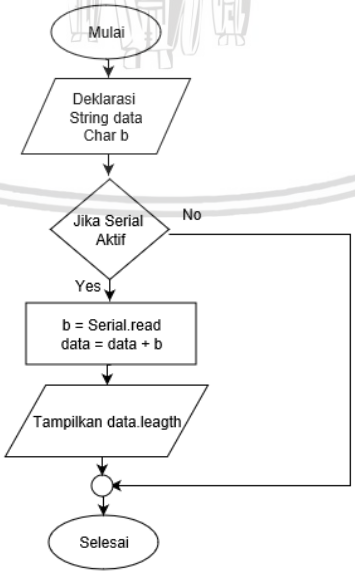
Pada gambar 5.10 diagram alir perancangan pengiriman data sensor dari arduino ke nodeMCU *sender,* proses pengiriman nilai data sensor dari arduino ke nodeMCU *sender* yaitu dengan menggunakan komunikasi serial.



Gambar 5.10 Diagram alir pengiriman data sensor dari arduino ke nodemcu *sender*

5.1.3.5 Perancangan penerimaan data sensor dari arduino ke nodemcu sender

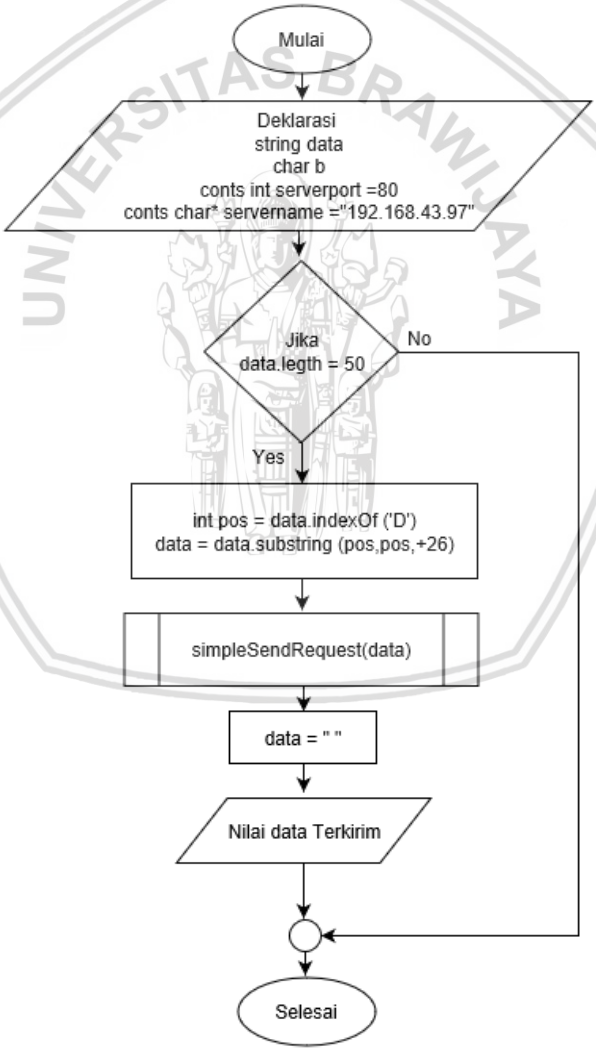
Proses penerimaan data sensor dari arduino ke nodeMCU *sender* yang ditunjukan pada gambar 5.11 dimulai dari pendeklarasian string data dan char b. Dalam sistem tersebut terdapat sebuah kondisi dimana, jika serial aktif maka variabel char b akan menampung data serial yang kemudian dimasukan pada variabel string data, setalah itu tampilkan jumlah karatkater pada variabel data. Dan jika serial tidak aktif maka proses selesai.



Gambar 5.11 Diagram alir pengiriman data sensor dari arduino ke nodemcu *sender*

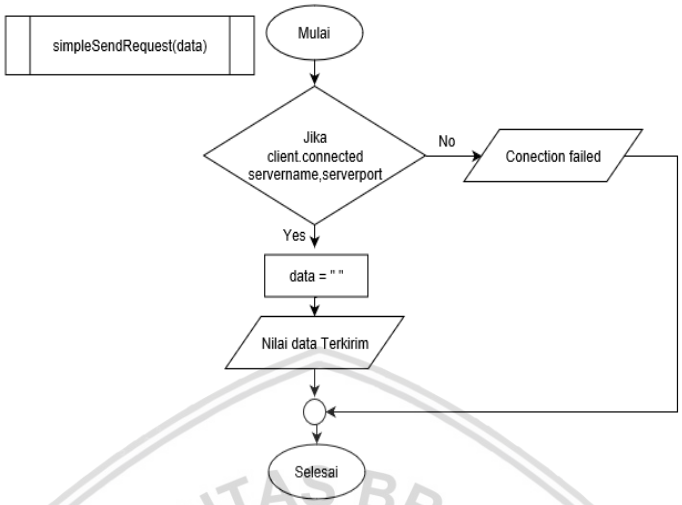
5.1.3.6 Perancangan pengiriman data sensor dari nodemcu sender ke nodemcu server

Proses pengiriman data sensor dari NodeMCU *sender* ke NodeMCU *server* yang dijuntukan pada gambar 5.12 yaitu diawalai dengan pendeklarasian string data, char b, conts int serverport = 80, dan conts char* servername 192.168.43.97. setelah pendeklarasian terdapat sebuah kondisi dimana, jika panjang karakter variabel data sama dengan 50 maka variabel pos dengan tipe data integer menyimpan nomer indeks karekter 'D' pada variabel data. Variabel data mengambil karakter yang dimulai dari huruf 'D' sampai 26 karakter selanjutnya. Setelah proses tersebut terdapat fungsi simpleSendRequest yang mempengaruhi pengiriman data sensor dari nodeMCU *sender* ke NodeMCU *server.* Adapun fungsi simpleRequest (data) yang ditunjukan dan dijelaskan pada gambar 5.13.



Gambar 5.12 Diagram alir perancangan pengiriman data sensor dari nodemcu *sender* ke nodemcu *server*

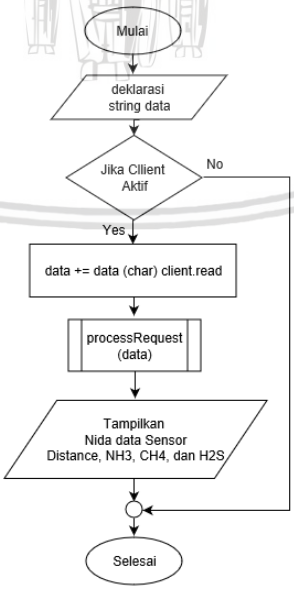
Pemanggilan fungsi simplesendrequest pada gambar 5.13 yaitu untuk mengirim variabel data menggunakan komunikasi TCP/Ip dan menampilkan data yang terkirim. Kemudian variabel data dikosongkan untuk diisi kembali.



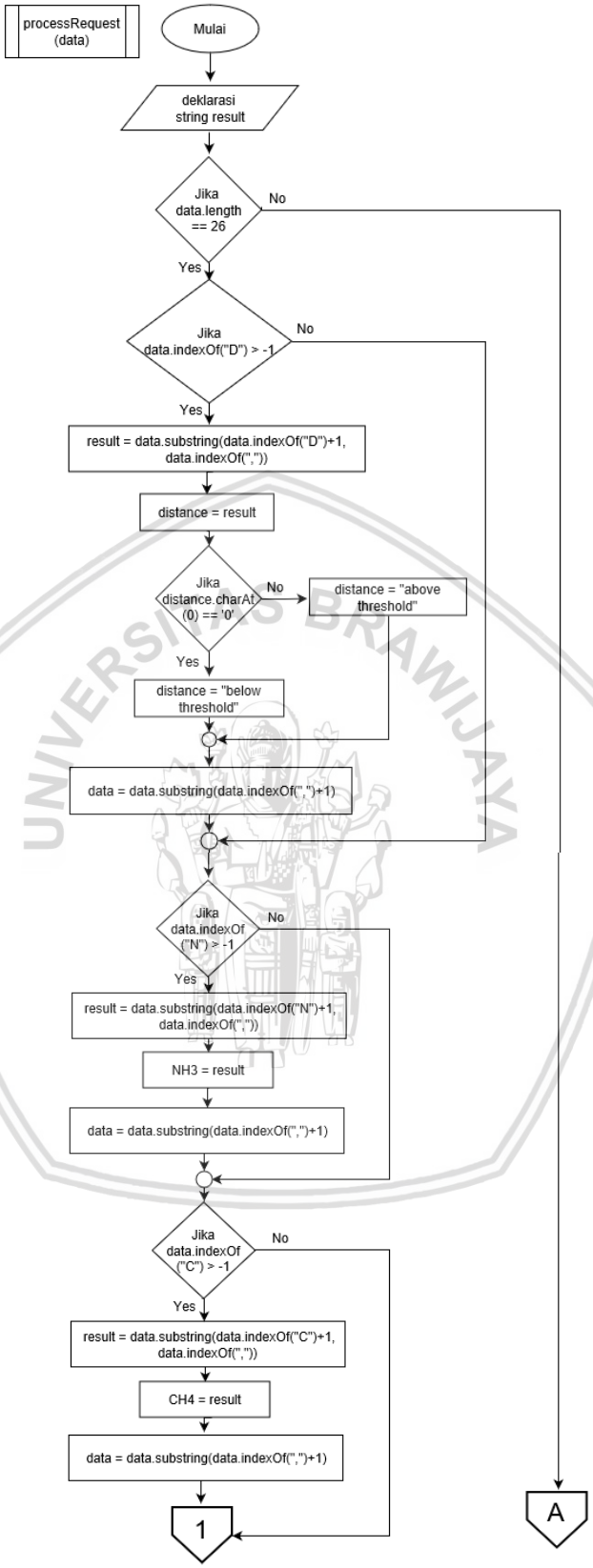
Gambar 5.13 Diagram alir fungsi simpleSendRequest ()

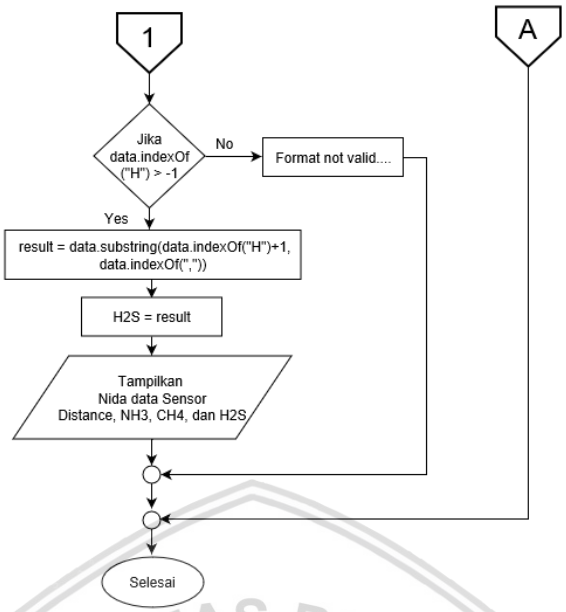
5.1.3.7 Perancangan penerimaan data sensor dari nodemcu sender ke nodemcu server

Pada gambar 5.14 proses penerimaan data sensor dari NodeMCU *sender* ke NodeMCU *server* ialah diawali dengan pendeklarasian string data yang kemudian terdapat sebuah kondisi dimana, jika client aktif maka masukan kiriman data *sender* ke variabel data. adapun penjelasan dari fungsi proses request (data) yang dijelaskan pada gambar 5.15.



Gambar 5.14 Diagram alir penerimaan data dari nodemcu *sender* ke nodemcu *server*


processRequest (data)
Mulai
deklarasi string result
Jika data.length == 26
No
Yes
Jika data.indexOf("D") > -1
No
Yes
result = data.substring(data.indexOf("D")+1, data.indexOf(","))
distance = result
Jika distance.charAt (0) == '0'
No
distance = "above threshold"
Yes
distance = "below threshold"
data = data.substring(data.indexOf(",")+1)
Jika data.indexOf ("N") > -1
No
Yes
result = data.substring(data.indexOf("N")+1, data.indexOf(","))
NH3 = result
data = data.substring(data.indexOf(",")+1)
Jika data.indexOf ("C") > -1
No
Yes
result = data.substring(data.indexOf("C")+1, data.indexOf(","))
CH4 = result
data = data.substring(data.indexOf(",")+1)
1
A

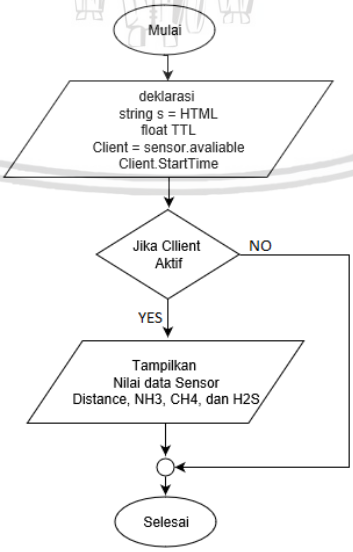


Gambar 5.15 Diagram alir FungsiRequeat ()

Pada gambar 5.15 diagram alir fungsi prosesRequest dimana pada proses ini akan menampilkan nilai dari sensor Infrared E18-D80NK yakni *"below threshold"* yang benilai 0 dan *"above threshold"* yang bernilai 1, nilai data sensor NH3,nilai data sensor CH4, dan nilai data sensor H2S.

5.1.3.8 Perancangan menampilkan data sensor pada website

Pada diagram alir perancangan Website 5.16, diawali pendeklarasian string s, yang mana didalam string s tersebut terdapat sebuah bahasa pemograman html yang digunakan untuk membuat halaman web yang berisikan nilai data sensor



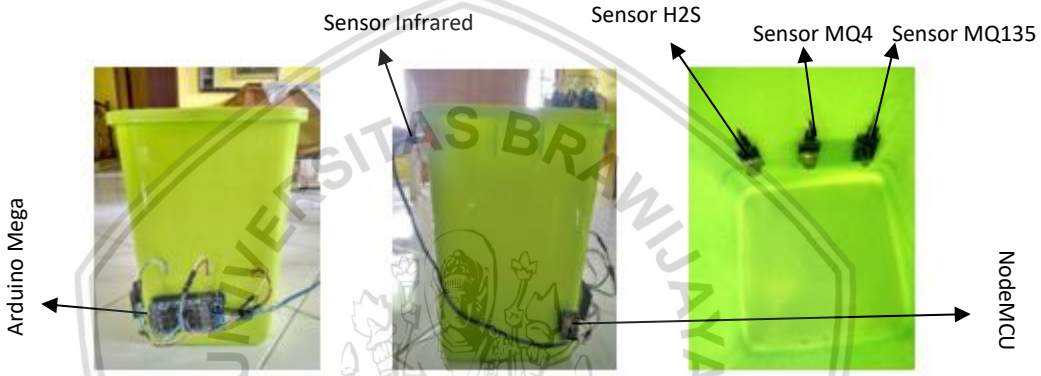
Gambar 5.16 Diagram alir perancangan menampilkan data sensor paada website

5.2 Implementasi sistem

Implementasi sistem yaitu merupakan suatu tahapan yang telah dilakukan sebelumnya berdasarkan semua perancangan agar pembuatan ssstem dapat terealisasi. Pada subbab ini implementasi *prototype*, implementasi perangkat keras serta implementtasi perangkat lunak akan dijelaskan satu persatu secara detail.

5.2.1 Implementasi *prototype* alat monitoring sampah

Untuk mengimplementasikan perancangan di subbab 5.1.1 yang berupa *prototype* sistem monitoring sampah yaitu menggunakan *prototype* tempat sampah dengan ukuran 22x22x33 cm. Maka hasil implementasi prototype dan peletakan komponen elektronik akan ditunjukkan pada Gambar 5.17 berikut.

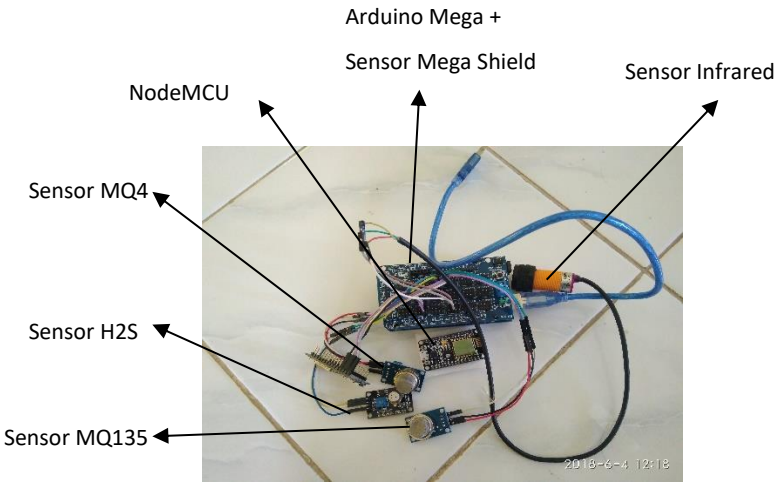


Gambar 5.17 Implementasi *Prototype* Sistem Trashmonitoring

Cara pertama atau kedua membantu pembaca yang ingin memisahkan observasi dan terjemahan dari observasi tersebut sehingga mereka dapat menilai kualitas dari masing-masing proses dengan lebih mudah. Kadang-kadang cara kedua lebih banyak dipilih daripada cara pertama jika data yang harus dipresentasikan yang cukup banyak dan laporan penelitian cukup panjang agar pembaca tidak perlu menunggu presentasi dari seluruh data selesai baru dapat membaca penerjemahannya. Cara pertama dan kedua ini banyak digunakan untuk penelitian yang bersifat kuantitatif, baik itu deskriptif, eksplanatori, maupun implementatif.

5.2.2 Implementasi perangkat keras

Implementasi Perangkat Keras ini akan menerangkan proses perangkat keras yang mencakup komponen elektronik antara lain Arduino mega, sensor gas TGS2602, sensor gas MQ-135, sensor gas MQ-4, sensor Infrared E18-D80NK E18-D80NK dan nodeMCU. Semua komponen dirangkai menjadi satu menggunakan mega sensor shield v 2.0 berdasarkan yang telah dijelaskan pada subbab 5.1.2.



Gambar 5.18 Implementasi rangkaian semua sensor dengan arduino mega

Pada gambar 5.18 adalah hasil implementasi ke-4 sensor serta nodeMCU *sender* dengan arduino mega, dimana sensor TGS2602, sensor MQ-4, dan sensor MQ-135 yang diletakkan sesuai perancangan yakni pas dibagian bawah dalam pada *prototype* tempat sampah, kemudian sensor infrared E18-D80NK diletakan di pojok atas temapt, sedangkan nodeMCU *sender* diletakkan dibagian samping tempat sampah bersama arduino mega yang terlihat pada gambar 5.18 bagian kanan yang dihubungkan melalui kabel jumper.

5.2.3 Implementasi perangkat lunak

Implementasi perangkat lunak menjelaskan proses realisasi program untuk Trashmonitoring berdasarkan perancangan yang telah dilakukan pada subbab 5.1.3. Dalam melakukan implementasi perangkat lunak ini sepenuhnya menggukana proses pengkodean program yang dilakukan pada Arduino mega IDE 1.6.4 dimana diawal program dilakukan inisialisasi library yang digunakan untuk mempermudah pemrograman beberapa fungsi tertentu. Pada Tabel 5.6 ditunjukkan pengimplementasian library pada sistem ini, diantaranya adalah *library* "FreeRTOS.h" untuk penjadwalan sensor yang akan diakuisis, *library* "ESP8266.h"*sender* untuk melakukan pengiriman data ke *server* dan *library* "ESP8266.h"*server* untuk menerima data dari *sender* kemudian diteruskan ke sebuah website yang diakses melalui browser.

Tabel 5.6 Kode sumber inisialisasi library sistem monitoring sampah

Baris	Eksekusi_*System*_Guna_RTOS
1	`#include <Arduino_FreeRTOS.h>`
2	`#include <ESP8266WwiFi.h>`
3	`#include <ESP8266WiFi.h>`

5.2.3.1 Implementasi kode sumber pembacaan data sensor infrared, MQ135, MQ4, dan TGS2602

Untuk melakukan implementasi pemebacaan nilai dari suatu sensor perlu dilakukan inisialisasi variabel dan konfigurasi pin sensor berdasarkan perancangan perangkat keras dan perancangan perangkat lunak yang telah dijelaskan pada subbab sebelum-sebelumnya. Tabel 5.7 menunjukkan pada baris ke-1 sampai baris ke-8 ialah inisialisasi pin H2S, CH4, NH3, dan Distance. Baris ke-9 hingga baris

ke-10 adalah inisialisai ID pengiriman dan total waktu eksekusi task dan baris ke-12 hingga baris ke-22 adalah inisialisasi nilai konstan sensor NH3, CH4, H2S yang dibutuhkan untuk mendapatkan nilai PPM dari tiap sensor.

Tabel 5.7 Kode sumber inisialisasi variabel dan pin sensor

Baris	Eksekusi_*System*_Guna_RTOS
1	`int H2SPin = A5; //TGS2602 pin`
2	`int CH4Pin = A4; //MQ-4 pin`
3	`int NH3Pin = A3; //MQ-135 pin`
4	`int distancePin = 5; //Infrared pin`
5	`static float H2SPin = 0; //TGS2602 value`
6	`static float CH4Pin = 0; //MQ4 value`
7	`static float NH3Pin = 0; //MQ135 value`
8	`static float distanceValue = 0; //Infrared value`
9	`int ID;`
10	`int wm;`
11	
12	`const int gasSensor = 0; // NH3`
13	`#define Ro 22000`
14	`float MQ135_SCALINGFACTOR = 37.58805473; //for NH3`
15	`float MQ135_EXPONENT = -3.235365807; //for NH3`
16	
17	`float TGS2602_SCALINGFACTOR = 7.05566582614; // for H2S`
18	`float TGS2602_EXPONENT = -2.954075758; // for H2S`
19	
20	`float m = -0.318; //Slope`
21	`float b = 1.133; //Y-Intercept`
22	`float R0 = 11.820; //Sensor Resistance in fresh air from previous code`

Implementasi kode sumber pembacaan nilai sensor infrared E18-D80NK dalam mendeteteksi ketinggian dalam sampah yang dideklarasikan pada fungsi DistanceTask () yang ditunjukan pada tabel 5.8 dimana pada baris ke-3 hingga baris ke-7 adalah proses pengambilan *semaphore,* selanjutnya pada baris ke-8 hingga baris ke-14 yaitu proses menampilkan ID dan total waktu eksekusi dalam pembacaan seluruh task sensor, kemudian pada baris ke-14 hingga baris ke-20 adalah proses perolehan nilai distance yang didapatkan dari dari nilai digital yang dibaca oleh sensor, nilai dari sensor infrared ini bernilai 1 dan 0, dimana nilai 1 " below Threshold" sedangkan 0 "above Threshold".

Tabel 5.8 Kode sumber pembacaan nilai sensor Infrared E18-D80NK

Baris	Eksekusi_*System*_Guna_RTOS
1	`static void DistanceTask(void* pvParameters)`
2	`{`
3	`for (;;)`
4	`{`
5	`  if ( xSemaphoreTake( xSerialSemaphore, ( TickType_t ) 1000 )`
6	`== pdTRUE )`
7	`{`
8	`ID++;`
9	`    Serial.print("==================");`
10	`    Serial.print("ID=");`
11	`    Serial.println(ID);`
12	`    wm = micros() ;`
13	`    Serial.println(wm);`
14	`    Serial.print(F("Distance measure !"));`

```
15      //read
16      distanceValue = digitalRead(distancePin);
17      //send
18      Serial.println(distanceValue);
19      Serial3.print("D");
20      Serial3.print(distanceValue);
21      Serial3.print(",");
22         xSemaphoreGive( xSerialSemaphore );
23      }
24       vTaskDelay(1000 / portTICK_PERIOD_MS );
25    }
26  }
```

Implementasi kode sumber pembacaan nilai sensor MQ135 dalam mendeteteksi kadar gas amonia dalam sampah yang dideklarasikan pada fungsi NH3Task () yang ditunjukan pada tabel 5.9 dimana pada baris ke-3 hingga baris ke-7 adalah proses pengambilan *semaphore,* yang kemudian pada baris ke-8 hingga baris ke-21 adalah proses perolehan nilai sensor MQ135 dari kadar gas amonia yang dihasilkan oleh sampah yang dideteksi. Sensor MQ135 adalah sensor resistif dimana nilai PPM amonia didapatkan dari nilai analog yang dibaca oleh sensor, yang kemudian diubah menjadi digital dengan proses ADC. Nilai digital ini pula yang menjadi tegangan masukan pada mikrokontroler arduino. Selanjutnya untuk mengetahui nilai resistansi yang dihasilkan oleh sensor yaitu dengan cara menggunakan rumus pembagi tegangan yang ditunjukkan pada baris ke-13. Sehingga setelah mengetahui nilai resistansi yang dihasilkan sensor dapat dilakukan kalibrasi untuk mendapatkan nilai PPM yang dimaksud dengan cara yang ditunjukkan pada baris ke-14.

Tabel 5.9 Kode sumber pembacaan nilai sensor MQ-135 (NH3)

Baris	Eksekusi_*System*_Guna_RTOS
	TrashMonitoring ClientRTOSv2
1	static void NH3Task(void* pvParameters)
2	{
3	for (;;)
4	{
5	if (xSemaphoreTake(xSerialSemaphore, (TickType_t) 1000
6	) == pdTRUE)
7	
8	Serial.print(F("NH3 measure !"));
9	//read
10	//send
11	float analog = analogRead(NH3Pin);
12	float voltage = (analog/1024)*5;
13	float Rs = ((Ro / voltage * 5) - Ro);
14	NH3Value = MQ135_SCALINGFACTOR * pow((Rs / Ro),
	MQ135_EXPONENT);
15	Serial.println(NH3Value);
16	Serial3.print("N");
17	Serial3.print(NH3Value);
18	Serial3.print(",");
19	xSemaphoreGive(xSerialSemaphore);
20	vTaskDelay(1000/portTICK_PERIOD_MS);
21	}

Implementasi kode sumber pembacaan nilai sensor MQ4 dalam mendeteteksi kadar gas amonia dalam sampah yang dideklarasikan pada fungsi CH4Task () yang

ditunjukan pada tabel 5.10 dimana pada baris ke-3 hingga baris ke-7 adalah proses pengambilan *semaphore,* yang kemudian pada baris ke-8 hingga baris ke-22 adalah proses perolehan nilai sensor MQ4 dari kadar gas metan yang dihasilkan oleh sampah yang dideteksi. Sensor MQ4 adalah sensor resistif dimana nilai PPM metana didapatkan dari nilai analog yang dibaca oleh sensor, yang kemudian diubah menjadi digital dengan proses ADC. Nilai digital ini pula yang menjadi tegangan masukan pada mikrokontroler arduino. Selanjutnya untuk mengetahui nilai resistansi yang dihasilkan oleh sensor yaitu dengan cara menggunakan rumus pembagi tegangan yang ditunjukkan pada baris ke-16. Sehingga setelah mengetahui nilai resistansi yang dihasilkan sensor dapat dilakukan kalibrasi untuk mendapatkan nilai PPM yang dimaksud dengan cara yang ditunjukkan pada baris ke-21.

Tabel 5.10 Kode sumber pembacaan nilai sensor MQ-4 (CH4)

Baris	Eksekusi_*System*_Guna_RTOS
	TrashMonitoring ClientRTOSv2
1	static void CH4Task(void* pvParameters)
2	{
3	for (;;)
4	{
5	if (xSemaphoreTake(xSerialSemaphore, (TickType_t) 1000
6	) == pdTRUE)
7	{
8	Serial.print(F("CH4 measure !"));
9	//read
10	CH4Value = analogRead(CH4Pin);
11	float sensor_volt; //Define variable for sensor voltage
12	float RS_gas; //Define variable for sensor resistance
13	float ratio; //Define variable for ratio
14	sensor_volt = CH4Value * (5.0 / 1023.0); //Convert analog
15	values to voltage
16	RS_gas = ((5.0 * 10.0) / sensor_volt) - 10.0; //Get value of
17	RS in a gas
18	ratio = RS_gas / R0; // Get ratio RS_gas/RS_air
19	double ppm_log = (log10(ratio) - b) / m; //Get ppm value in
20	linear scale according to the the ratio value
21	double ppm = pow(10, ppm_log); //Convert ppm value to log
22	scale
23	//send
24	Serial.println(CH4Value);
25	Serial3.print("C");
26	Serial3.print(CH4Value);
27	Serial3.print(","); xSemaphoreGive(xSerialSemaphore);
28	vTaskDelay(1000/portTICK_PERIOD_MS);
29	}

Implementasi kode sumber pembacaan nilai sensor TGS2602 dalam mendeteteksi kadar gas amonia dalam sampah yang dideklarasikan pada fungsi H2STask () yang ditunjukan pada tabel 5.11 dimana pada baris ke-3 hingga baris ke-7 adalah proses pengambilan *semaphore,* yang kemudian pada baris ke-8 hingga baris ke-29 adalah proses perolehan nilai sensor TGS2602 dari kadar gas hidrogen sulfida yang dihasilkan oleh sampah yang dideteksi. Sensor TGS2602 adalah sensor resistif dimana nilai PPM hidrogen sulfida didapatkan dari nilai analog yang dibaca oleh sensor, yang kemudian diubah menjadi digital dengan proses ADC. Nilai digital ini pula yang menjadi tegangan masukan pada

mikrokontroler arduino. Selanjutnya untuk mengetahui nilai resistansi yang dihasilkan oleh sensor yaitu dengan cara menggunakan rumus pembagi tegangan yang ditunjukkan pada baris ke-13. Sehingga setelah mengetahui nilai resistansi yang dihasilkan sensor dapat dilakukan kalibrasi untuk mendapatkan nilai PPM yang dimaksud dengan cara yang ditunjukkan pada baris ke-14.

Tabel 5.11 Kode Sumber Pembacaan Nilai Sensor Gas TGS2602 (H2S)

Baris	Eksekusi_*System*_Guna_RTOS
	TrashMonitoring ClientRTOSv2
1	static void H2STask(void* pvParameters)
2	{
3	for (;;)
4	{
5	if (xSemaphoreTake(xSerialSemaphore, (TickType_t) 1000
6	) == pdTRUE)
7	{
8	Serial.print(F("H2S measure !"));
9	//read
10	H2SValue = analogRead(H2SPin);
11	float analog = analogRead(H2SPin);
12	float voltage = (analog/1024)*5;
13	float Rs = ((Ro / voltage * 5) - Ro);
14	H2SValue = TGS2602_SCALINGFACTOR * pow((Rs / Ro),
	TGS2602_EXPONENT);
15	//send
16	Serial.println(H2SValue);
17	Serial3.print("H");
18	Serial3.print(H2SValue);
19	Serial3.print(",");
20	int wd = micros() ;
21	Serial.print("Total Waktu Eksekusi :");
23	Serial.println(wd-wm); xSemaphoreGive(xSerialSemaphore);
24	vTaskDelay(1000/portTICK_PERIOD_MS);
25	}

5.2.3.2 Implementasi kode sumber RTOS

Pada bagian void setup() pendeklarasian fungsi *Semaphore* yang bertugas sebagai sumber daya atau *resource*. Dimana fungsi *Semaphore* ini akan dijalankan terlebih dahulu. Pertama dilakukan pengecekan pada *Semaphore,* jika *Semaphore* kosong maka *Semaphore* akan dibuat. Dan jika *Semaphore* tidak kosong maka *Semaphore* dapat digunakan. Program pembuatan *Semaphore* dapat lihat pada Tabel 5.12

Tabel 5.12 Kode sumber pembuatan *semaphore*

Baris	Eksekusi_*System*_Guna_RTOS
1	if (xSerial*Semaphore* == NULL)
2	
3	{
4	xSerial*Semaphore* = x*Semaphore*CreateMutex();
5	
6	if ((xSerial*Semaphore*) != NULL)
7	x*Semaphore*Give((xSerial*Semaphore*)); }
8	

Pada tabel 5.13 diatas adalah kode program RTOS dimana pada kode program baris ke-1 ialah sebuah fungsi untuk menjalakan *setup-setup* yang ada di program,

baris ke-2 dan ke-3 adalah inisialisasi nilai bautrate pada serial, baris ke-5 hingga baris ke-10 adalah sebuah kode sumber pembuatan task, stack size, dan prioritas pada tiap sensor dengan periotas yang sama yaitu 2 agar pengiriman nilai data dari masing-masing sensor terjadwal dan realtime.

Tabel 5.13 Kode Sumber Pembacaan RTOS

Baris	Eksekusi_*System*_Guna_RTOS
	TrashMonitoring ClientRTOSv2
1	void setup() {
2	// Deklarasi bautrate
3	Serial.begin(9600);
4	// Penggunakaan RTOS
5	xTaskCreate(DistanceTask, "MyDistanceTask", 256, NULL, 2,
6	NULL);
7	xTaskCreate(NH3Task, "MyNH3Task", 256, NULL, 2, NULL);
8	xTaskCreate(CH4Task, "MyCH4Task", 256, NULL, 2, NULL);
9	xTaskCreate(H2STask, "MyH2STask", 256, NULL, 2, NULL);
10	}

5.2.3.3 Implementasi kode sumber koneksi wifi

Pada tabel 5.14 kode sumber koneksi wifi baris ke-1 hingga baris ke-3 adalah pendeklarasian nama wifi yakni : “Redmi” dan pasword dari dari wifi tersebut ialah “88888888”. Setelah pendeklarasian nama dan pasword pada diagram alir tersebut terdapat fungsi initWiFi () Dimana didalam fungsi initWiFi terdapat sebuah kondisi, jika Status WiFi terhubung maka akan menampilkan Local IP, dan jika status WiFi tidak terhubung maka akan menampilkan “.”.

Tabel 5.14 Kode Sumber Koneksi WiFi

Baris	TrashMonitoring *Server*
1	#include <ESP8266WiFi.h>
2	const char* SSIDName = "Redmi";
3	const char* SSIDPass = "8888888888";
4	void setup() {
5	initWiFiSTA();
6	Serial.print("Local IP : ");
7	Serial.println(WiFi.localIP());
8	}
9	
10	void loop() {
11	// put your main code here, to run repeatedly:
12	
13	}

5.2.3.4 Implementasi kode sumber pengiriman data sensor dari mikrokontroler arduino ke nodemcu sender

Tabel 5.15 kode sumber pengiriman data sensor dari arduino ke nodeMCU *sender,* pada baris ke-1 terdapat fungsi setup () baris ke-2 inisialisasi nilai boutrate pada serial, dan baris ke-4 hingga baris ke-19 adalah proses pengiriman data sensor distance, NH3, CH4, dan H2S dengan menggunakan komunikasi serial.

Tabel 5.15 Kode Sumber Pengiriman Data Sensor Dari Arduino Ke NodeMCU *sender*

Baris	TrashMonitoring *Sender*
1	`void setup() {`
2	`  Serial3.begin(9600);`
3	`}`
4	`//Pengiriman data sensor inframera`
5	`    Serial3.print("D");`
6	`    Serial3.print(distanceValue);`
7	`    Serial3.print(",");`
8	`//Pengiriman data sensor MQ-135`
9	`    Serial3.print("N");`
10	`    Serial3.print(NH3Value);`
11	`    Serial3.print(",");`
12	`//Pengiriman data sensor MQ-4`
13	`    Serial3.print("C");`
14	`    Serial3.print(CH4Value);`
15	`    Serial3.print(",");`
16	`//Pengiriman data sensor TGS2602`
17	`    Serial.println(H2SValue);`
18	`    Serial3.print("H");`
19	`    Serial3.print(H2SValue);`

5.2.3.5 Implementasi kode sumber penerimaan data sensor dari mikrokontroler arduino ke nodemcu sender

Proses penerimaan data sensor dari arduino ke nodeMCU *sender* yang terlihat pada gamabar 5.16, dimulai dari pendeklarasian string data dan char b. Dalam sistem tersebut terdapat sebuah perulangan dan kondisi dimana, jika serial aktif maka variabel char b akan menampung data serial yang kemudian dimasukkan pada variabel string data, setalah itu menampilkan jumlah karatkter pada variabel data. Dan jika serial tidak aktif maka proses selesai.

Tabel 5.16 Kode Sumber Penerimaan Data Sensor Dari Arduino Ke NodeMCU *Sender*

Baris	TrashMonitoring *Sender*
	`//deklarasi variabel menerima data dari arduino`
1	`String data;`
2	`char b;`
3	`void loop() {`
4	`  if(Serial.available()){`
5	`    b=Serial.read();`
6	`    data = data + b;`
7	`    //Serial.print(data);`
8	`    Serial.println(data.length());`
9	`  }`
10	`}`

5.2.3.6 Implementasi kode sumber pengiriman data sensor dari nodemcu sender ke nodemcu server

Proses pengiriman data sensor dari NodeMCU *sender* ke NodeMCU *server* yang dijuntukan pada tabel 5.17 yaitu diawali dengan pendeklarasian string data, char b, conts int serverport = 80, dan conts char* servername 192.168.43.97 yang

terletak pada baris ke-1 hingga baris ke-6. Setelah pendeklarasian terdapat sebuah perulangan didalmanya terdapat sebuah kondisi dimana, jika panjang karakter variabel data sama dengan 50 maka variabel pos dengan tipe data integer menyimpan nomer indeks karekter 'D' pada variabel data. Variabel data mengambil karakter yang dimulai dari huruf 'D' sampai 27 karakter selanjutnya. Setelah proses tersebut terdapat fungsi simpleSendRequest untuk mengirim variabel data menggunakan komunikasi TCP/Ip dan menampilkan data yang terkirim. Setelah itu variabel data dikosongkan untuk diisi kembali.

Tabel 5.17 Kode Sumber Pengiriman Data Sensor nodeMCU *Server* Ke *NodeMCU Server*

Baris	TrashMonitoring *Server*
	TrashMonitoringSender
1	const int serverPort = 80;
2	const char* serverName = "192.168.43.97";
3	const int TTL = 5000;
4	WiFiClient client;
5	
6	void setup() {
7	Serial.begin(9600);
8	}
9	
10	void loop() {
11	if(data.length()==50){
12	int pos = data.indexOf('D');
13	data = data.substring(pos,pos+27);
14	simpleSendRequest(data);
15	data="";
16	Serial.print("Terkirim ");
17	Serial.print(data);
18	}
19	}
20	
21	void simpleSendRequest(String request){
22	Serial.print("Request : ");
23	Serial.println(request);
24	// Use WiFiClient class to create TCP connections
25	if (!client.connect(serverName, serverPort)) {
26	Serial.println("Connection failed");
27	return;
28	}
29	// This will send the request to the server
30	client.print(request);
31	}

5.2.3.7 Implementasi kode sumber penerimaan data sensor dari nodemcu sender ke nodemcu server

Proses penerimaan data sensor dari NodeMCU *sender* ke NodeMCU *server* pada gambar 5.18, diawali dengan pendeklarasian string data pada baris ke-4, didalam perulangan terdapat sebuah kondisi dimana, jika client aktif maka masukan kiriman data *sender* ke variabel data. Kemudian setelah proses tersebut terdapat fungsi prosesRequest (data) dimana jika panjang data sama dengan 27 maka server akan menampilkan data sensor Infrared E18-D80NK, CH4, NH3, dan H2S dengan cara indexOf dan substring.

Tabel 5.18 Kode Sumber Penerimaan Data Sensor NodeMCU *Sender* Ke NodeMCU *Server*

Baris	TrashMonitoring Server
	TrashMonitoringServer
1	void loop() {
2	Serial.println("");
3	Serial.println("Request: ");
4	String data = "";
5	while(client.available()){
6	data += (char)client.read();
7	}
8	processRequest(data);
9	Serial.println(data);
10	}
11	void processRequest(String data){
12	String result;
13	if(data.length() == 27){
14	Serial.println("Processing ...");
15	if(data.indexOf("D") > -1){
16	result = data.substring(data.indexOf("D")+1,
17	data.indexOf(","));
18	distance = result;
19	if(distance.charAt(0) == '0'){
20	distance = "above threshold";
21	}else{
22	distance = "below threshold";
23	}
24	Serial.println(result);
25	data = data.substring(data.indexOf(",")+1);
26	}
27	if(data.indexOf("N") > -1){
28	result = data.substring(data.indexOf("N")+1,
29	data.indexOf(","));
30	NH3 = result;
31	Serial.println(result);
32	data = data.substring(data.indexOf(",")+1);
34	}
35	if(data.indexOf("C") > -1){
36	result = data.substring(data.indexOf("C")+1,
37	data.indexOf(","));
38	CH4 = result;
39	Serial.println(result);
40	data = data.substring(data.indexOf(",")+1);
41	}
42	if(data.indexOf("H") > -1){
43	result = data.substring(data.indexOf("H")+1);
44	data.indexOf(","));
45	H2S = result;
46	Serial.println(result);
47	}
48	}else{
49	Serial.println("Format not valid ...");
50	}
51	}

5.2.3.8 Implementasi kode sumber menampilkan data sensor pada website

Pada tabel 5.19 kode sumber website menampilkan data sensor, didalam perulangan baris ke-6 hingga baris ke-30 tedapat pendeklarasian string s, yang mana didalam string s tersebut terdapat sebuah bahasa pemograman html yang

digunakan untuk membuat halaman web yang berisikan nilai data sensor Infrared, Ch4, NH3, dan H2S. Dan baris ke-32 hingga baris ke-48 adalah kode sumber untuk cek koneksi.

Tabel 5.19 Kode sumber menampilkan data sensor pada website

Baris	TrashMonitoring *Server*
	TrashMonitoringServer
1	void setup() {
2	// put your setup code here, to run once:
3	
4	}
5	
6	void loop() {
7	String s = "HTTP/1.1 200 OK\r\n";
8	s += "Content-Type: text/html\r\n\r\n";
9	s += "<!DOCTYPE HTML>\r\n";
10	s += "<html>\r\n";
11	s += " <head>\r\n";
12	s += " <title>Sensor Monitoring</title>\r\n";
13	s += " </head>\r\n";
14	s += " <body>\r\n";
15	s += " <table>\r\n";
16	s += " <tr>\r\n";
17	s += " <th>Distance</th>\r\n";
18	s += " <th>NH3</th>\r\n";
19	s += " <th>CH4</th>\r\n";
20	s += " <th>H2S</th>\r\n";
21	s += " </tr>\r\n";
22	s += " <tr>\r\n";
23	s += " <td>" + distance + "</td>\r\n";
24	s += " <td>" + NH3 + "</td>\r\n";
25	s += " <td>" + CH4 + "</td>\r\n";
26	s += " <td>" + H2S + "</td>\r\n";
27	s += " </tr>\r\n";
28	s += " </table>\r\n";
29	s += " </body>\r\n";
30	s += "</html>\n";
31	// Cek Koneksi
32	float TTL = 100000;
33	client = server.available();
34	client.setTimeout(TTL);
35	IPAddress clientIP = client.localIP();
36	if (!client) {
37	return false;
38	}
39	float startTime = millis();
40	while(!client.available()){
41	if(millis() - startTime < TTL){
42	Serial.print(".");
43	delay(250);
44	}else{
45	Serial.print("Timeout!");
46	return false;
47	}
48	}
49	client.print(s);
50	client.flush();

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian serta menganalisis hasil dari pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Pengujian dilakukan bertujuan untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenui oleh sistem. Proses pengujian yang dilakukan adalah berupa pengujian fungsional, pengujian akurasi dan pengujian pengiriman data sistem, dimana pengujian fungsional yakni menguji fungsi dari perangkat keras dalam hal ini berupa sensor-sensor yang digunakan serta pengiriman data mengunakan NodeMCU apakah dapat bekerja sesuai dengan program yang telah dirancang, pengujian akurasi yakni menguji seberapa akurat sistem yang telah dirancang dan diimplementasikan, sedangkan pengujian pengiriman data yakni menguji apakah data sensor yang telah di dideteksi dapat terkirim dengan baik sesuai dengan penschadule sensor yang telah dirancang sehingga data sensor tersebut dapat dikrimkan pada sebuah website sederhana.

6.1 Pengujian sensor gas MQ135

Sensor gas MQ-135 adalah sensor yang memiliki peranan dan tugas dalam melakukan proses pembacaan nilai terhadap kadar gas amonia pada sampah yang dideteksi. Pada pengujian ini proses pengukuran nilai dari kadar gas amonia dilakukan secara mandiri melalui sisa sisa makanan, kotoran hewan, dan bangkai tumbahan yang telah membusuk.

6.1.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini adalah untuk menerapkan dan mengetahui fungsionalitas sensor gas MQ-135 dalam pembacaan gas amonia apakah sensor MQ-135 ini sesuai dengan karakteristik pada datasheet.

6.1.2 Prosedur pengujian

Berikut prosedur yang dilakukan untuk menguji sensor gas MQ-135 :

1. Menghubungkan mikrokontroller Arduino Mega dengan laptop.
2. Meng-upload kode program dari sensor MQ-135.
3. Mengukur nilai sensor MQ-135 melalui serial monitor dalam 10 detik, untuk pembacaan nilai awal tanpa gas amonia , kadar gas amonia pada sampah dan kadar gas menggunakan gas amonia.
4. Mengamati dan menganalisis hasil pembacaan nilai dari sensor MQ-135 yang di hasilkan oleh sampah, cairan gas amonia, dan keadaan kosong.
5. Kesimpulan

6.1.3 Hasil dan analisis pengujian

Tabel 6.1 menunjukkan nilai pembacaan sensor MQ-135 pada pengujian kadar gas amonia. Dengan menggunakan 3 kondisi yaitu pertama ketika tempat sampah

dalam keadaan kosong, kedua menggunakna caiaran gas amonia,dan yang ketiga menggunakan sampah busuk. Berdasarkan pengujian pada tabel 6.1 dari 10 kali pengujian terlihat bahwa nilai sensor MQ-135 terjadi perubahan. Sehingga dapat disimpulkan bahwa sensor MQ135 dapat bekerja dengan baik. Hal tersebut dapat dilihat dari hasil pengamatan data uji sampah, cairan gas amonia dan tanpa cairan gas amonia pada tempat sampah menunjukan berubahan nilai data sensor. Dari 10 kali pengujian nilai rata-rata kadar gas amonia pada sensor MQ-135 sebesar 35.71 PPM. Hal ini menunjukan kadar gas amonia pada sampah tersebut tidak terlalu banyak.

Tabel 6.1 Hasil pengujian pembacaan sensor MQ135

No	OUTPUT Kondisi sampah Kosong	OUTPUT Cairan Gas Amonia	OUTPUT Sampah Busuk	Status
1	6.12 PPM	224.74 PPM	30.5 PPM	Terjadi Perubahan
2	6.12 PPM	82.61 PPM	30.7 PPM	Terjadi Perubahan
3	6.03 PPM	144.43 PPM	30.7 PPM	Terjadi Perubahan
4	6.12 PPM	85.85 PPM	35.7 PPM	Terjadi Perubahan
5	6.12 PPM	53.57 PPM	35.3 PPM	Terjadi Perubahan
6	6.03 PPM	69.96 PPM	37.2 PPM	Terjadi Perubahan
7	6.03 PPM	88.09 PPM	37.3 PPM	Terjadi Perubahan
8	6.03 PPM	92.74 PPM	37.8 PPM	Terjadi Perubahan
9	5.12 PPM	96.40 PPM	40.4 PPM	Terjadi Perubahan
10	5.12 PPM	77.62 PPM	41.5 PPM	Terjadi Perubahan
Average	5.884 PPM	101.601 PPM	35.71 PPM	Terjadi Perubahan

6.2 Pengujian sensor gas MQ4

Sensor gas MQ-4 adalah sensor yang memiliki peranan dan tugas dalam melakukan proses pembacaan nilai kadar gas metana pada sampah yang dideteksi. Pada pengujian ini proses pengukuran nilai dari kadar gas metana dilakukan secara mandiri melalui sisa sisa makanan, kotoran hewan, dan bangkai tumbahan yang telah membusuk.

6.2.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini iyalah untuk menerapkan dan mengetahui fungsionalitas sensor gas MQ-4 dalam pembacaan gas metana apakah sensor MQ-4 ini sesuai dengan karakteristik pada datasheet.

6.2.2 Prosedur pengujian

Berikut prosedur yang dilakukan untuk menguji sensor gas MQ4 :

1. Menghubungkan mikrokontroler Arduino Mega dengan laptop.
2. Meng-upload kode program dari sensor MQ-4.
3. Mengukur nilai sensor MQ-4 melalui serial monitor dalam 10 detik, untuk pembacaan nilai awal tanpa gas metan , kadar gas metan pada sampah dan kadar gas menggunakan gas metan.
4. Mengamati dan menganalisis hasil pembacaan nilai dari sensor MQ-4 yang di hasilkan oleh sampah, gas metan, dan keadaan kosong.
5. Kesimpulan.

6.2.3 Hasil dan analisis pengujian

Tabel 6.2 menunjukkan nilai pembacaan sensor MQ4 pada pengujian kadar gas metana. Dengan menggunakan 3 kondisi yaitu pertama ketika tempat sampah dalam keadaan kosong, kedua menggunakna caiaran gas metana,dan yang ketiga menggunakan sampah busuk. Berdasarkan pengujian pada tabel 6.2 dari 10 kali pengujian terlihat bahwa nilai sensor MQ4 terjadi perubahan. Sehingga dapat disimpulkan bahwa sensor MQ4 dapat bekerja dengan baik. Hal tersebut dapat dilihat dari hasil pengamatan data uji sampah, cairan gas metana dan tanpa cairan gas metana pada tempat sampah menunjukan berubahan nilai data sensor. Dari 10 kali pengujian nilai rata-rata kadar gas metana pada sensor MQ4 sebesar 293.5 PPM. Hal ini menunjukan kadar gas metana pada sampah tersebut cukup banyak.

Tabel 6.2 Hasil pengujian pembacaan sensor MQ-4

No	OUTPUT Kondisi sampah Kosong	OUTPUT Cairan Gas Metana	OUTPUT Sampah Busuk	Status
1	256 PPM	354 PPM	280 PPM	Terjadi Perubahan
2	255 PPM	340 PPM	285 PPM	Terjadi Perubahan
3	255 PPM	375 PPM	290 PPM	Terjadi Perubahan
4	255 PPM	441 PPM	290 PPM	Terjadi Perubahan
5	255 PPM	497 PPM	290 PPM	Terjadi Perubahan

No	OUTPUT Kondisi sampah Kosong	OUTPUT Cairan Gas Metana	OUTPUT Sampah Busuk	Status
6	255 PPM	427 PPM	295 PPM	Terjadi Perubahan
7	255 PPM	447 PPM	300 PPM	Terjadi Perubahan
8	255 PPM	810 PPM	300 PPM	Terjadi Perubahan
9	255 PPM	824 PPM	300 PPM	Terjadi Perubahan
10	256 PPM	835 PPM	305 PPM	Terjadi Perubahan
Average	255.2 PPM	535 PPM	293.5 PPM	Terjadi Perubahan

6.3 Pengujian sensor gas TGS2602

Sensor gas TGS2602 adalah sensor yang memiliki peranan dan tugas dalam melakukan proses pembacaan nilai terhadap kadar gas hidrogen sulfida pada sampah yang dideteksi. Pada pengujian ini proses pengukuran nilai dari kadar gas hidrogen sulfida dilakukan secara mandiri melalui telur busuk, sisa sisa makanan, kotoran hewan, dan bangkai tumbahan yang telah membusuk.

6.3.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini iyalah untuk menerapkan dan mengetahui fungsionalitas sensor gas TGS2602 dalam pembacaan gas hidrogen sulfida apakah sensor TGS2602 ini sesuai dengan karakteristik pada datasheet.

6.3.2 Prosedur pengujujian

Berikut prosedur yang dilakukan untuk menguji sensor gas TGS2602 :

1. Menghubungkan mikrokontroler Arduino Mega dengan laptop.
2. Meng-upload kode program dari sensor TGS2602.
3. Mengukur nilai sensor TGS2602 melalui serial monitor dalam 10 detik, untuk pembacaan nilai awal tanpa gas hidrogen sulfida , kadar gas hidrogen sulfida pada sampah dan kadar gas menggunakan gas hidrogen sulfida.
4. Mengamati dan menganalisis hasil pembacaan nilai dari sensor TGS2602 yang di hasilkan oleh sampah, gas hidrogen sulfida, dan keadaan kosong.
5. Kesimpulan.

6.3.3 Hasil dan analisis pengujian

Tabel 6.3 menunjukkan nilai pembacaan sensor TGS2602 pada pengujian kadar gas metana. Dengan menggunakan 3 kondisi yaitu pertama ketika tempat sampah

dalam keadaan kosong, kedua menggunakna caiaran gas hidrogen sulfida, dan yang ketiga menggunakan sampah busuk. Berdasarkan pengujian pada tabel 6.3 dari 10 kali pengujian terlihat bahwa nilai sensor TGS2602 terjadi perubahan. Sehingga dapat disimpulkan bahwa sensor TGS2602 dapat bekerja dengan baik. Hal tersebut dapat dilihat dari hasil pengamatan data uji sampah, cairan gas hidrogen sulfida dan tanpa cairan gas hidrogen sulfida pada tempat sampah menunjukan berubahan nilai data sensor. Dari 10 kali pengujian nilai rata-rata kadar gas hidrogen sulfida pada sensor TGS2602 sebesar 9.738 PPM. Hal ini menunjukan kadar gas hidrogen sulfida pada sampah tidak terlalu banyak.

Tabel 6.3 Hasil pengujian pembacaan sensor TGS2602

No	OUTPUT Kondisi sampah Kosong	OUTPUT Cairan Gas Hidrogen sulfida	OUTPUT Sampah Busuk	Status
1	0.15 PPM	23.26 PPM	09.10 PPM	Terjadi Perubahan
2	0.19 PPM	22.44 PPM	08.12 PPM	Terjadi Perubahan
3	0.18 PPM	13.34 PPM	09.23 PPM	Terjadi Perubahan
4	0.21 PPM	11.74 PPM	10.09 PPM	Terjadi Perubahan
5	0.21 PPM	14.64 PPM	10.15 PPM	Terjadi Perubahan
6	0.19 PPM	19.91 PPM	07.02 PPM	Terjadi Perubahan
7	0.17 PPM	16.67 PPM	11.11 PPM	Terjadi Perubahan
8	0.15 PPM	12.44 PPM	10.85 PPM	Terjadi Perubahan
9	0.11 PPM	13.03 PPM	11.01 PPM	Terjadi Perubahan
10	0.21 PPM	13.66 PPM	10.70 PPM	Terjadi Perubahan
Average	0.177 PPM	16.113 PPM	9.738 PPM	Terjadi Perubahan

6.4 Pengujian sensor Infrared

Sensor Infrared E18-D80NK adalah sensor yang memiliki peranan dan tugas dalam melakukan proses pembacaan nilai ketinggian terhadap volume tempat sampah melalui tinggi sampah yang dideteksi.

6.4.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini iyalah untuk menerapkan dan mengetahui fungsionalitas sensor Infrared E18-D80NK dalam pembacaan volume tempat sampah melalui tinggi sampah yang dideteksi apakah sensor Infrared E18-D80NK

ini sesuai dengan karakteristik pada datasheet. Pembacaan sensor Infrared E18-D80NK ini diukur dari letak sensor dan permukaan sampah.

6.4.2 Prosedur pengujian

Berikut prosedur yang dilakukan untuk menguji sensor Infrared E18-D80NK :

1. Menghubungkan mikrokontroler Arduino Mega dengan laptop.
2. Meng-upload kode program dari sensor Infrared E18-D80NK.
3. Mengukur nilai dari sensor Infrared E18-D80NK untuk pembacaan volume sampah melalui tingggi sampah pada tempat sampah dengan 5x.
4. Mengamati dan menganalisis hasil pembacaan nilai dari sensor Infrared E18-D80NK yang diukur dari volume samapah melalui tinggi sampah pada tempat sampah.
5. Kesimpulan.

6.4.3 Hasil dan analisis pengujian

Pada Tabel 6.4 menunjukkan nilai pembacaan sensor infrared E18-D80NK dengan 5x percobaan dimana percobaan pertama sampah tersebut diletakkan di dasar *prototype* tempat sampah dengan volume yang tidak begitu penuh sehingga output dari nilai sensor tersebut adalah below threshold, dan pada pecobaan kedua sampah diletakkan hingga menyentuh sensor infrared E18-D80NK sehingga output dari nilai sensor tersebut above thereshold, sama halnya dengan pengujian ke-3, ke-4, dan ke-5 yang dapat dilihat pada tabel 6.4. oleh karena itu dapat disimpulkan bahwa sensor infrared dapat bekerja dengan baik.

Tabel 6.4 Hasil Pengujian Pembacaan Sensor E18-D80NK

No	Nama Pengujian	Output Sensor infrared	Keterangan
1	Pengujian ke-1	Below Threshold	Sampah tidak melebihi batas tempat sampah
2	Pengujian ke-2	Above Threshold	Sampah melebihi batas tempat sampah
3	Pengujian ke-3	Below Threshold	Sampah tidak melebihi batas tempat sampah
4	Pengujian ke-4	Above Threshold	Sampah melebihi batas tempat sampah
5	Pengujian ke-5	Above Threshold	Sampah melebihi batas tempat sampah

6.5 Pengujian RTOS

6.5.1 Pengujian eksekusi task berdasarkan prioritas

6.5.1.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui sistem penjadwalan dari semua sensor apakah sensor-sensor tersebut dapat berjalan sesuai dengan program. Hal ini perlu dilakukan agar mengetahui apakah sensor-sensor tersebut dapat di akuisis dengan baik oleh mikrokontroller menggunakan RTOS atau tidak.

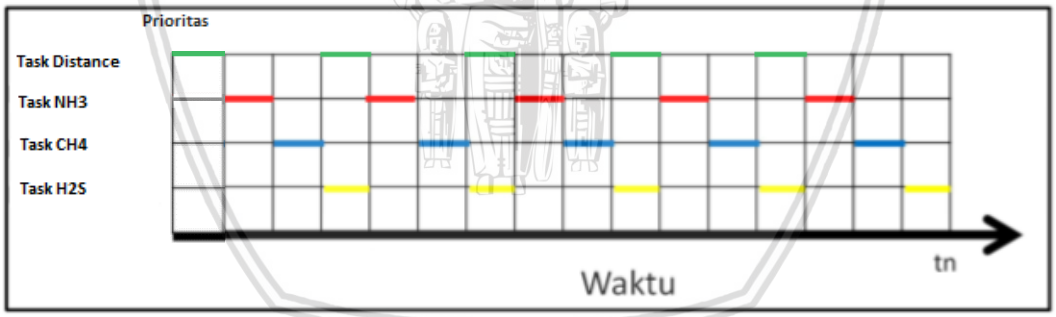
6.5.1.2 Prosedur pengujian

Untuk melakukan prosedur pengujian ini yaitu dengan cara ;

1. Menghububungkan prototipe alat dengan komputer atau laptop.
2. Buka Software Arduino IDE, setelah itu buka source code alat yang dibuat kemudian compile dan upload source code programnya.
3. Amati hasil pengujian eksekusi task berdasarkan prioritas melalui serial monitor.
4. Kesimulan.

6.5.1.3 Hasil dan analisis pengujian

Hasil dari pengujian eksekusi *task* berdasarkan proritas yang dilakukan melalui serial monitor, dapat dilihat pada Gambar 6.1



Gambar 6.1 Analisis hasil pengujian eksekusi *task* berdasarkan *prioritas*

Berdasarkan analisis Gambar 6.1, *Task* sensor *Distance*, NH3, CH4, dan H2S akan dieksekusi terlebih dahulu, hal ini dikarenakan task *distance,* NH3, CH4, dan H2S memiliki *prioritas* yang sama yaitu 2, dimana *prioritas* tersbut merupakan *prioritas* terbesar dalam sistem. Selama *task* 2 dieksekusi maka *task* dibawahnya akan menunggu di eksekusi hingga *task* 2 menyelesaikan tugasnya, hal ini dikarenakan terdapat fungsi *Semaphore*() pada sistem yang bertugas agar *task* tersebut tidak dapat di *interrupt* oleh *task* lain selama *task* tersebut masih dieksekusi**.**

6.5.2 Pengujian waktu eksekusi task pada sistem

6.5.2.1 Tujuan pengujian

Tujuan pengujian ini adalah untuk memastikan waktu eksekusi setiap *task* pada program tidak melebihi deadline yang telah ditentukan yaitu 1000 ms.

6.5.2.2 Prosedur pengujian

Untuk melakukan prosedur pengujian ini yaitu dengan cara ;

1. Menghubungkan prototipe alat dengan komputer atau laptop.
2. Buka Software Arduino IDE, setelah itu buka source code alat yang dibuat kemudian compile dan upload source code programnya.
3. Amati hasil pengujian waktu eksekusi task melalui serial monitor.
4. Kesimpulan.

6.5.2.3 Hasil dan analisis pengujian

Hasil pengujian waktu eksekusi tiap *task* pada sistem yang telah ditentukan deadlinenya dapat diliat pada Tabel 6.5.

Tabel 6.5 Analisis hasil pengujian waktu eksekusi tiap *task*

Task	**Waktu Eksekusi**	***Deadline***	**Hasil**
Task Distance	±1.1 ms	1000 ms	Tidak Melebihi *Deadline*
Task NH3	±1.3 ms		Tidak Melebihi *Deadline*
Task CH4	±1.3 ms		Tidak Melebihi *Deadline*
Task H2S	±1.3 ms		Tidak Melebihi *Deadline*
Total Waktu	**±5.2 ms**		Tidak Melebihi *Deadline*

Berdasarkan hasil analisis yang terdapat pada tabel 6.5 dapat diketahui bahwa waktu eksekusi dari setiap *task* tidak melebihi *deadline* waktu yang sudah ditentukan yaitu 1000 ms. *Task Distance* memiliki waktu eksekusi lebih kecil dari *task* NH3, CH4, dan H2S, hal ini di karenakan pada *task Distance* hanya memiliki sebuah kondisi logika 1 dan 0, sehingga tidak memerlukan waktu yang banyak.

6.5.3 Pengujian perbandingan waktu eksekusi sistem monitoring sampah menggunakan RTOS dengan tanpa menggunakan RTOS

6.5.3.1 Tujuan pengujian

Tujuan dari pengujian ini adalah membandingkan kecepatan eksekusi *task* pada sistem yang menggunakan RTOS dengan sistem tanpa menggunakan RTOS.

6.5.3.2 Prosedur pengujian

Untuk melakukan prosedur pengujian ini yaitu dengan cara ;

1. Menghubungkan prototipe alat dengan komputer atau laptop.
2. Buka Software Arduino IDE, setelah itu buka source code alat yang dibuat kemudian compile dan upload source code programnya.
3. Amati hasil pengujian perbandingan waktu eksekusi sistem monitoring dengan RTOS dan tanpa RTOS melalui serial monitor.
4. Kesimpulan.

6.5.3.3 Hasik dan analisis pengujian

Pengujian perbandingan waktu eksekusi sistem monitoring dengan RTOS dan tanpa mengunakan RTOS pada tabel 6.6 menunjukan bahwa proses eksekusi menggunakan RTOS lebih cepat dibandingkan dengan tanpa menggunakan RTOS. Hal ini dapat dilihat dari rata-rata selisih waktu eksekusi dari sistem yaitu 27.420 ms. Proses eksekusi RTOS lebih cepat dibandingkan dengan tanpa mengunakan RTOS dikarenakan sistem operasi pada RTOS mengeksekusi *task*-task pada sistem secara bersamaan dengan menggunakan konsep konkurensi.

Tabel 6.6 Hasil Pebandingan Waktu Eksekusi Dengan RTOS Dan Tanpa RTOS

Sampel	Waktu eksekusi dengan RTOS (ms)	Waktu eksekusi Tanpa RTOS(ms)
1	2.224	25.528
2	2.224	30.548
3	2.220	30.548
4	2.220	30.548
5	2.216	30.548
6	2.212	30.548
7	2.212	29.516
8	2.220	29.516
9	2.184	29.516
10	2.188	29.508
Average	2.212	29.632

6.6 Pengujian pengiriman data sensor dari mikrokontroler arduino ke nodemcu sender

6.6.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui apakah data hasil akusisi sensor yang dilakukan oleh mikrokontroller dengan menggunakna RTOS ke NodeMCU *sender* dapat terkirim dengan baik.

6.6.2 Prosedur pengujian

Berikut ini prosedur pengujian proses pengiriman data sensor dari arduino ke nodeMCU *sender*:

1. Menghubungkan prototipe alat dengan komputer atau laptop.
2. Buka Software Arduino IDE, setelah itu buka source code alat yang dibuat setelah itu compile dan upload source code programnya.
3. Amati hasil proses pengiriman data sensor dari arduino ke NodeMCU sender melalui serial monitor.
4. Kesimpulan

6.6.3 Hasil dan analisis pengujian

Berdasarkan hasil pengujiaan pengiriman data dari arduino ke nodeMCU *sender* pada tabel 6.7 menunjukan bahwa paket data sensor terkirim dengan baik, meskipun paket data tersebut terkadang tidak sesuai dengan dengan urutan pengiriman dari arduino, hal ini dikarenakan pada nodeMCU *sender* masih melakukan proses koneksi wifi yang mengakibatkan beberapa paket data yang dikirimkan oleh arduino tidak dapat terbaca.

Tabel 6.7 Hasil Dan Analisis Pengiriman Data Dari Arduino Ke NodeMCU *Sender*

Data	Sensor	Output pada arduino mega	Output pada nodemcu *sender*	Status
Pengiriman ke 1	Distance	1.00	1.00	Terkirim
	NH3	224.74 PPM	224.74 PPM	Terkirim
	CH4	354 PPM	354 PPM	Terkirim
	H2S	09.10 PPM	09.10 PPM	Terkirim
Pengiriman ke 2	Distance	1.00	1.00	Terkirim
	NH3	82,61 PPM	82,61 PPM	Terkirim
	CH4	340 PPM	340 PPM	Terkirim
	H2S	08.12 PPM	08.12 PPM	Terkirim
Pengiriman ke 3	Distance	1.00	1.00	Terkirim
	NH3	144,43 PPM	144,43 PPM	Terkirim
	CH4	375 PPM	375 PPM	Terkirim
	H2S	09.23 PPM	09.23 PPM	Terkirim
Pengiriman ke 4	Distance	0.00	0.00	Terkirim
	NH3	85,85 PPM	85,85 PPM	Terkirim

Data	Sensor	Output pada arduino mega	Output pada nodemcu *sender*	Status
	CH4	441 PPM	497 PPM	Terkirim
	H2S	10.09 PPM	10.09	Terkirim
Pengiriman ke 5	Distance	0.00	0.00	Terkirim
	NH3	53,57 PPM	53,57 PPM	Terkirim
	CH4	497 PPM	427 PPM	Terkirim
	H2S	10.15 PPM	10.15 PPM	Terkirim
Pengiriman ke 6	Distance	1.00	1.00	Terkirim
	NH3	69,96 PPM	69,96	Terkirim
	CH4	427 PPM	447 PPM	Terkirim
	H2S	07.02 PPM	07.02 PPM	Terkirim
Pengiriman ke 7	Distance	0.00	0.00	Terkirim
	NH3	88,09 PPM	88,09 PPM	Terkirim
	CH4	447 PPM	447 PPM	Terkirim
	H2S	11.11 PPM	11.11 PPM	Terkirim
Pengiriman ke 8	Distance	0.00	0.00	Terkirim
	NH3	92,74 PPM	92,74 PPM	Terkirim
	CH4	810 PPM	810 PPM	Terkirim
	H2S	10.85 PPM	10.85 PPM	Terkirim
Pengiriman ke 9	Distance	0.00	0.00	Terkirim
	NH3	96,40 PPM	96,40 PPM	Terkirim
	CH4	824 PPM	824 PPM	Terkirim
	H2S	11.01 PPM	11.01 PPM	Terkirim
Pengiriman ke 10	Distance	0.00	0.00	Terkirim
	NH3	77,62 PPM	77,62 PPM	Terkirim
	CH4	835 PPM	835 PPM	Terkirim
	H2S	10.70 PPM	10.70 PPM	Terkirim

6.7 Pengujian pengiriman data sensor dari nodemcu sender ke nodemcu server

6.7.1 Tujuan pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui apakah data hasil akusisi sensor yang dilakukan oleh NodeMCU *sender* ke NodeMCU *server* dapat terkirim dengan baik.

6.7.2 Prosedur pengujian

Berikut ini prosedur pengujian proses pengiriman data sensor dari NodeMCU *sender* ke NodeMCU *server* :

1. Menghubungkan prototipe alat dengan komputer atau laptop.
2. Buka Software Arduino IDE, setelah itu buka source code alat yang dibuat setelah itu compile dan upload source code programnya.
3. Amati hasil proses pengiriman data sensor dari NodeMCU *sender* ke NodeMCU *server* melalui serial monitor.
4. Kesimpulan

6.7.3 Hasil dan analisis pengujian

Berdasarkan hasil pengujian pengiriman data dari nodeMCU *sender* ke nodeMCU *server* pada tabel 6.8 menunjukan bahwa paket data yang dikirimkan oleh nodeMCU *sender* ke nodeMCU *server* 100 % terkirim secara akurat.

Tabel 6.8 Hasil Dan Analisis Pengiriman Data Dari NodeMCU *Sender* Ke NodeMCU *Server*

Data	Sensor	Output pada nodemcu *sender*	Nodemcu *server*	Status
Pengiriman ke 1	Distance	1.00	1.00	Terkirim
	NH3	224.74 PPM	224.74 PPM	Terkirim
	CH4	354 PPM	354 PPM	Terkirim
	H2S	09.10 PPM	09.10 PPM	Terkirim
Pengiriman ke 2	Distance	1.00	1.00	Terkirim
	NH3	82,61 PPM	82,61 PPM	Terkirim
	CH4	340 PPM	340 PPM	Terkirim
	H2S	08.12 PPM	08.12 PPM	Terkirim
Pengiriman ke 3	Distance	1.00	1.00	Terkirim
	NH3	144,43 PPM	144,43 PPM	Terkirim
	CH4	375 PPM	375 PPM	Terkirim

Data	Sensor	Output pada nodemcu *sender*	Nodemcu *server*	Status
	H2S	09.23 PPM	09.23 PPM	Terkirim
Pengiriman ke 4	Distance	0.00	0.00	Terkirim
	NH3	85,85 PPM	85,85	Terkirim
	CH4	441 PPM	441 PPM	Terkirim
	H2S	10.09 PPM	10.09 PPM	Terkirim
Pengiriman ke 5	Distance	0.00	0.00	Terkirim
	NH3	53,57 PPM	53,57 PPM	Terkirim
	CH4	497 PPM	497 PPM	Terkirim
	H2S	10.15 PPM	10.15 PPM	Terkirim
Pengiriman ke 6	Distance	1.00	1.00	Terkirim
	NH3	69,96 PPM	69,96 PPM	Terkirim
	CH4	427 PPM	427 PPM	Terkirim
	H2S	07.02 PPM	07.02 PPM	Terkirim
Pengiriman ke 7	Distance	0.00	0.00	Terkirim
	NH3	88,09 PPM	88,09 PPM	Terkirim
	CH4	447 PPM	447 PPM	Terkirim
	H2S	07.02 PPM	07.02 PPM	Terkirim
Pengiriman ke 8	Distance	0.00	0.00	Terkirim
	NH3	92,74 PPM	92,74 PPM	Terkirim
	CH4	810 PPM	810 PPM	Terkirim
	H2S	10.85 PPM	10.85 PPM	Terkirim
Pengiriman ke 9	Distance	0.00	0.00	Terkirim
	NH3	96,40 PPM	96,40 PPM	Terkirim
	CH4	824 PPM	824 PPM	Terkirim
	H2S	11.01 PPM	11.01 PPM	Terkirim
Pengiriman ke 10	Distance	0.00	0.00	Terkirim
	NH3	77,62 PPM	77,62 PPM	Terkirim
	CH4	835 PPM	835 PPM	Terkirim
	H2S	10.70 PPM	10.70 PPM	Terkirim

BAB 7 PENUTUP

7.1 Kesimpulan

Sesuai dengan rumusan masalah yang diajukan diawal penelitian serta berdasarkan hasil analisis dari pengujian yang dilakukan maka dapat ditarik beberapapa kesimpulan sebagai berikut :

1. Sensor infrared, sensor gas MQ-135, MQ-4, dan TGS2602 yang diletakkan dibagian *prototype* tempat sampah yang berwarna hijau dapat membaca kadar gas dan volume ketinggian tempat sampah dengan baik. Sensor MQ-135 dengan 10x percobaan dapat membaca kadar gas amonia dengan rata-rata 35.71 PPM , sensor MQ-4 dengan 10x percobaan dapat membaca kadar gas metana rata-rata 293.5 PPM dan sensor TGS2602 dapat membaca kadar gas hidrogen sulfida dengan 10x percobaan memperoleh rata-rata 9.738 PPM. Jadi kadar gas yang dihasilkan oleh sampah lebih didominasi oleh gas amonia dan gas metana.
2. Pengimplementasian *Real Time Operating System* pada mikrokontroler dapat berjalan dengan baik, pertama terbukti dengan adanya pengujian eksekusi tiap task berdasarkan prioritas dapat tereksekusi semua, kedua pengujian waktu eksekusi task pada sistem tidak melebihi deadline yang diberikan dan yang ketiga pengujian perbandingan waktu eksekusi sistem monitoring dengan RTOS dan tanpa RTOS ternyata lebih cepat menggunakan RTOS dengan rata-rata waktu eksekusi 2.212 ms dengan selisih 27.420 ms dari pengujian tanpa menggunakan RTOS. Hali ini dikarenakan sistem operasi pada RTOS mengeksekusi *task*-task pada sistem secara bersamaan dengan menggunakan konsep konkurensi.
3. Performa sistem dalam mengamati kondisi tempat sampah yang diukur dari pengiriman data 100 % terkirim dengan akurat. Terbukti dengan terkirimnya data sensor dari mikrokontroler nodemcu sender ke nodemcu server yang sama persis dan data sensor yang dapat diakses melalui alat penampil pada website.

7.2 Saran

Beberapa saran yang dapat dijadikan untuk pengembangan penelitian yang serupa kedepannya yakni berikut:

1. Menggunakan metode atau algoritma untuk menentukan sebuah pengambilan keputusan.
2. Mengenbangkan desain tampilan website agar terlihat menarik
3. Menerapkan sistem serupa dengan tampilan antarmuka melalui mobile apps sehingga lebih memudahkan mobilitas penggunaannya.
4. Mengunakan sensor yang lebih khusus untuk mendeteksi kadar gas amonia dan kadar gas metana agar lebih meningkatkan keakuratan pembacaan sensor.

DAFTAR PUSTAKA

Agung Leona Suparlin, S. R. D. S., 2017. *Implementasi System Real Time Untuk Monitoring Pencahayaan Suhu dan Kelembaban pada tanaman Stroberi.* malang: Universitas Brawijaya.

Albastaki, Y. F., 2018. *Electronic Nose Technologies and Advances in Machine Olfaction.* Hershey PA,USA: IGI Global.

Almuchlisin, A. N. J. U. A. A., 2016. *PERANCANGAN DAN IMPLEMENTASI SISTEM MONITORING UNTUK PELAPORAN SAMPAH BERBASIS TEKNOLOGI EMBEDDED.* s.l.:Universitas Telkom.

Aynur Unal, M. N. K. M. S. J., 2016. *Smart Trends in Information Technology and Computer Communications.* Singapore: Springer Nature.

Carl K. Chang, L. C. C. J. M. A., 2016. *Inclusive Smart Cities and Digital Health.* Switzerland: Springer Internasional.

Durfee, W., 2011. Arduino Microcontroller Guide. pp. 1-27.

Liam Downey, M. V. W., 2011. The Mental Health Impacts of Living Near Industrial Activity. *HHS Author Manuscripts,* pp. 1-7.

Radm Robert C. Williams, P. D., 2001. *Landfill Gas Primer.* s.l.:Agency for Toxic Substances and Disease Registry.

Schwastz, M., 2017. *ESP8266 Internet Of Thing Cockbook.* Bringmingham: packt Publising Ltd.

willard, W., 2006. 5 penyunt. s.l.:s.n.

Wisnu Jatmiko, P. M. ,. J., 2015. *TEORI DAN APLIKASI REAL TIME OPERATING SYSTEM (RTOS).* Depok: Universitas Indonesia.